



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS141501

**RANCANG BANGUN AGREGATOR TOKO APLIKASI
MOBILE BERBASIS WEB UNTUK MENYEDIAKAN
INFORMASI KOMPATIBILITAS APLIKASI MULTI
PLATFORM**

***DEVELOPMENT OF WEB-BASED MOBILE
APPLICATION STORE AGGREGATOR TO PROVIDE
COMPATIBILITY INFORMATION OF MULTI
PLATFORM APPLICATIONS***

**PRI REZKI DESTRIANTO
NRP 5213 100 151**

Dosen Pembimbing
Rully Agus Hendrawan S.Kom., M.Eng.
Andre Parvian Aristio, S.Kom., M.Sc.

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KS141501

RANCANG BANGUN AGREGATOR TOKO APLIKASI MOBILE BERBASIS WEB UNTUK MENYEDIAKAN INFORMASI KOMPATIBILITAS APLIKASI MULTI PLATFORM

PRI REZKI DESTRIANTO
NRP 5213 100 151

Dosen Pembimbing
Rully Agus Hendrawan S.Kom., M.Eng.
Andre Parvian Aristio, S.Kom., M.Sc.

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

UNDERGRADUATE THESES - KS141501

**DEVELOPMENT OF WEB-BASED MOBILE
APPLICATION STORE AGGREGATOR TO PROVIDE
COMPATIBILITY INFORMATION OF MULTI
PLATFORM APPLICATIONS**

PRI REZKI DESTRIANTO
NRP 5213 100 151

Supervisor
Rully Agus Hendrawan S.Kom., M.Eng.
Andre Parvian Aristio, S.Kom., M.Sc.

INFORMATION SYSTEMS DEPARTMENT
Information Technology Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2017

Lembar Pengesahan

RANCANG BANGUN AGREGATOR TOKO APLIKASI MOBILE BERBASIS WEB UNTUK MENYEDIAKAN INFORMASI KOMPATIBILITAS APLIKASI MULTI PLATFORM

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

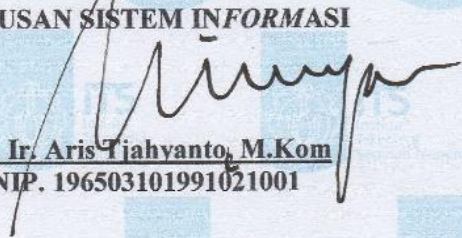
Departemen Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

Pri Rezki Destrianto
NRP. 5213100151

Surabaya, Juli 2017

KEPALA
JURUSAN SISTEM INFORMASI


Dr. Ir. Aris Tjahvanto, M.Kom
NIP. 196503101991021001

Lembar Persetujuan

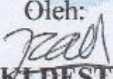
RANCANG BANGUN AGREGATOR TOKO APLIKASI MOBILE BERBASIS WEB UNTUK MENYEDIAKAN INFORMASI KOMPATIBILITAS APLIKASI MULTI PLATFORM

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada


Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:


PRI REZKIDESTRIANTO

NRP. 5213100151

Disetujui Tim Penguji : Tanggal Ujian : 07 Juli 2017
Periode Wisuda : September 2017

Rully Agus Hendrawan, S.Kom., M.Eng.  (Pembimbing I)

Andre Parvian Aristio, S.Kom., M.Eng.  (Pembimbing II)

Mahendrawati ER, ST, M.Sc, Ph.D  (Penguji I)

Erma Suryani, ST., MT., Ph.D.  (Penguji II)

RANCANG BANGUN AGREGATOR TOKO APLIKASI MOBILE BERBASIS WEB UNTUK MENYEDIAKAN INFORMASI KOMPATIBILITAS APLIKASI MULTI PLATFORM

Nama Mahasiswa : Pri Rezki Destrianto

NRP : 5213 100 151

Jurusan : Sistem Informasi FTIF-ITS

Pembimbing 1 : Rully Agus Hendrawan S.Kom., M.Eng.

Pembimbing 2 : Andre Parvian Aristio, S.Kom., M.Sc.

ABSTRAK

Layanan konten digital berkembang seiring meningkatnya jumlah aplikasi yang melimpah pada layanan konten digital. Para pengguna aplikasi menginginkan cara yang efektif dan efisien dalam melakukan evaluasi berbagai aplikasi yang tersedia dengan cepat dan sesuai dengan kebutuhannya masing-masing. Namun saat ini, tidak jarang pengguna aplikasi yang mengalami kesulitan memilih layanan konten yang tepat dalam mengevaluasi aplikasi tertentu. Sehingga pengguna memerlukan usaha lebih dalam menemukan informasi aplikasi yang ingin dievaluasi tersebut. Selain itu, keterbatasan media publikasi karya perangkat lunak saat ini, membatasi eksistensi dan penghargaan bagi Indie Developer.

Oleh karena itu dibutuhkan direktori perangkat lunak yang dapat menyajikan daftar perangkat lunak dari berbagai sumber layanan konten digital secara terintegrasi untuk memudahkan pengguna dalam mengevaluasi dan melakukan pencarian perangkat lunak secara efektif dan efisien. Teknik yang digunakan dalam penelitian ini yaitu dengan menggunakan agregasi detail informasi produk dari banyak toko aplikasi mobile yang multi platform kedalam satu media. Luaran yang ada dalam penelitian ini berupa sebuah prototype aplikasi berbasis website berbasis PHP/HTML yang mampu menyajikan

software direktori aplikasi multi platform yang terstruktur dengan metode prototyping.

Hasil yang didapat adalah fungsi levenshtein mencapai keakuratan 90.7% untuk mencari similaritas. Angka ini tidak cukup mencapai titik keakuratan yang dibutuhkan, sedangkan penggabungan antara fungsi levenshtein dan fungsi similar text mencapai keakuratan 94.5% sudah mencapai tingkat keakuratan dalam mencari persamaan aplikasi di play store dan app store. Sehingga, metode gabungan dua fungsi yang diimplementasikan ke aplikasi. Aplikasi yang dibuat diharapkan dapat membantu pengguna untuk mengevaluasi dan mendapatkan informasi aplikasi multi platform secara detail dan lengkap serta dapat menjadi bahan pertimbangan dalam pemilihan device di masa depan.

Kata Kunci: Product Aggregation, Software Directory, Software Development

DEVELOPMENT OF WEB-BASED MOBILE APPLICATION STORE AGGREGATOR TO PROVIDE COMPATIBILITY INFORMATION OF MULTI PLATFORM APPLICATIONS

Nama Mahasiswa : Pri Rezki Destrianto

NRP : 5213 100 151

Jurusan : Sistem Informasi FTIF-ITS

Pembimbing 1 : Rully Agus Hendrawan S.Kom., M.Eng.

Pembimbing 2 : Andre Parvian Aristio, S.Kom., M.Sc.

ABSTRACT

Digital content services are evolving as the number of applications in digital content services increases. The users of the application want an effective and efficient way to evaluate the various applications available quickly and according to their individual needs. But today, it's often for app users who have trouble choosing the right content service in evaluating a particular app. So users need more effort in finding the application information that want to be evaluated. In addition, the limitations of the media publication of the current software work, limiting the existence and appreciation for Indie Developer.

Therefore, it takes a software directory that can present integrated list of software from various sources of digital content services to facilitate users in evaluating and searching software effectively and efficiently. The technique used in this research is by using aggregation of product information detail from multiplatform mobile application store into one media. The outcomes of this research is a prototype based application of website based on PHP / HTML that able to present multiplatform application directory software that is structured by prototyping method. The expected results in this study are to help users to evaluate and obtain multiplatform application information in detail and complete and can be a consideration in the selection of devices in the future.

The result that obtained in this study is levenshtein function reached 90.7% accuracy for similarity. This number does not quite reach the required point of accuracy, while the combination between levenshtein function and similar text function reaches 94.5% accuracy, reaching the level of accuracy in finding the equation of applications in the Google Play and App Store. So, the combination method of two functions that implemented in the application. The application is expected to help users to evaluate and gaining information in multiplatform application in detail and complete and can be used for consideration in choosing the device in the future.

Keywords: Product Aggregation, Software Directory, Software Development

KATA PENGANTAR

Alhamdulillah atas karunia, rahmat, barakah, dan jalan yang telah diberikan Allah SWT selama ini sehingga penulis mendapatkan kelancaran dalam menyelesaikan tugas akhir dengan judul:

RANCANG BANGUN AGREGATOR TOKO APLIKASI MOBILE BERBASIS WEB UNTUK MENYEDIAKAN INFORMASI KOMPATIBILITAS APLIKASI MULTI PLATFORM

Terima kasih atas pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik materi maupun spiritual demi tercapainya tujuan pembuatan tugas akhir ini. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sedalam-dalamnya kepada:

1. Bapak Dr. Ir. Aris Tjahyanto, M.Kom selaku Ketua Jurusan Sistem Informasi ITS Surabaya
2. Bapak Rully Agus Hendrawan S.Kom., M.Eng. dan Bapak Andre Parvian Aristio, S.Kom., M.Sc. selaku dosen pembimbing yang meluangkan waktu, memberikan ilmu, petunjuk, dan motivasi untuk kelancaran tugas akhir ini.
3. Ibu Mahendrawathi Er., ST., M.Sc., Ph.D. dan Ibu Erma Suryani ST., MT., Ph.D. selaku dosen penguji yang telah memberikan masukan untuk perbaikan tugas akhir ini.
4. Orang tua penulis, Sunarto dan Siti Supriatin yang telah mendokan dan mendukung dalam pengerjaan tugas akhir ini
5. Seluruh dosen Jurusan Sistem Informasi ITS yang telah memberikan ilmu yang sangat berharga bagi penulis.
6. Adimas Eka Putra, Rani Oktavia, Marina Safitri, Fitri Larasati, Harun Rizal, Stezar Priansya, dan teman-teman lab ADDI dan SE yang telah mendukung dan membantu menyelesaikan penulisan ini.
7. Rekan-rekan BELTRANIS yang telah berjuang bersama dalam menjalani perkuliahan di Jurusan Sistem Informasi ITS.

8. Berbagai pihak yang membantu dalam penyusunan Tugas Akhir ini dan belum dapat disebutkan satu per satu dengan dukungan, semangat, dan kebersamaan.

Penyusunan laporan ini masih jauh dari sempurna, untuk itu saya menerima adanya kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga buku tugas akhir ini dapat memberikan manfaat pembaca.

Surabaya, Juli 2017
Penulis,

(Pri Rezki Destrianto)

DAFTAR ISI

Lembar Pengesahan	vii
1 Lembar Persetujuan.....	ix
ABSTRAK.....	v
<i>ABSTRACT</i>	vii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xvi
DAFTAR TABEL.....	xix
1. BAB I PENDAHULUAN	1
1.1 Latar belakang	1
1.2 Rumusan masalah	3
1.3 Batasan masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	4
1.5.1 Bagi Konsumen	4
1.5.2 Bagi Developer Produk Perangkat Lunak	4
1.6 Relevansi	4
2 BAB II TINJAUAN PUSTAKA	7
2.1 Penelitian sebelumnya	7
2.1 Landasan teori.....	9
2.2.1 Google play store API	10
2.2.2 App store API	12
2.2.3 Laravel	13
2.2.4 Usability testing	14
2.2.5 Levenshtein Distance	14
2.2.6 Web Scraping	15

3	BAB III METODOLOGI PENELITIAN	17
3.1	Diagram Metodologi.....	17
3.2	Uraian Metodologi.....	18
3.2.1	Studi Literatur	18
3.2.2	Analisis Kebutuhan	19
3.2.3	Pengambilan Data	19
3.2.4	Perancangan Prototipe	19
3.2.5	Pengujian Aplikasi	20
3.2.6	Penyusunan Buku Tugas Akhir	20
3.3	Jadwal Kegiatan.....	20
4	BAB IV PERANCANGAN.....	23
4.1	Pengambilan Data	23
4.1.1	Perancangan Pengambilan Data	23
4.1.2	Desain Basis Data	29
4.2	Rancang Bangun Prototipe.....	64
4.2.1	<i>Requirement Gathering</i>	64
4.2.2	Antar Muka Pengguna	65
4.2.3	Use Case	69
4.2.4	<i>Descriptive Case</i>	70
5	BAB V IMPLEMENTASI	73
5.1	Perangkat Penelitian.....	73
5.2	Struktur Direktori.....	74
5.3	<i>Deployment</i> Website API.....	74
5.4	Pengambilan Data	75
5.4.1	Pembuatan Scraper Play Store	76
5.4.2	Pembuatan Scraper App Store	80
5.5	Perhitungan Similarity	81

5.6	Rancang Bangun Prototipe.....	83
5.6.1	Implementasi Halaman Daftar Aplikasi	84
5.6.2	Implementasi Halaman Detail Aplikasi.....	86
5.6.3	Implementasi Halaman Ulasan Aplikasi	88
5.6.4	Implementasi Halaman Login dan Register ..	90
5.6.5	Implementasi Halaman Masukan Data Aplikasi	90
6	BAB VI HASIL DAN PEMBAHASAN	93
6.1	Data aplikasi	93
6.2	Hasil Prototipe Aplikasi	95
6.2.1	Halaman Daftar Aplikasi	95
6.2.2	Halaman Detail Aplikasi	96
6.2.3	Halaman Ulasan Aplikasi	96
6.2.4	Halaman Masukan Data Secara Rinci	97
6.2.5	Halaman Masukan Data Berdasarkan Appid	98
6.2.6	Halaman <i>login</i>	98
6.2.7	Halaman Registrasi.....	99
6.3	Hasil Analisis perbandingan	99
6.3.1	Perbandingan Antara Levenshtein Dengan Gabungan Levenshtein dan Similar Text	99
6.4	Usability Testing.....	100
7	BAB VII KESIMPULAN DAN SARAN.....	103
7.1	Kesimpulan.....	103
7.2	Saran.....	104
	DAFTAR PUSTAKA	105
	BIODATA PENULIS	107
	Lampiran A	- 1 -

Lampiran B	- 1 -
Lampiran C	- 1 -
Lampiran D	- 1 -
Lampiran E.....	- 1 -
Lampiran F.....	- 1 -

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

GAMBAR 3.1 METODOLOGI PENELITIAN	18
GAMBAR 4.1 ALUR PENGAMBILAN DATA SEBELUM DIMODIFIKASI DI PLAY STORE	24
GAMBAR 4.2 ALUR PENGAMBILAN DATA SEBELUM DIMODIFIKASI DI APP STORE	24
GAMBAR 4.3 ALUR PENGAMBILAN DATA DARI API JSON.....	25
GAMBAR 4.4 SAMPEL DATA <i>LIST</i> PLAY STORE API	26
GAMBAR 4.5 SAMPEL DATA <i>APP</i> PLAY STORE API	26
GAMBAR 4.6 SAMPEL DATA <i>REVIEW</i> PLAY STORE API	27
GAMBAR 4.7 SAMPEL DATA <i>PERMISSION</i> PLAY STORE API.....	27
GAMBAR 4.8 SAMPEL DATA <i>LIST</i> APP STORE API	28
GAMBAR 4.9 SAMPEL DATA <i>APP</i> PADA APP STORE API.....	28
GAMBAR 4.10 ENTITAS APP STORE DAN PLAY STORE	30
GAMBAR 4.11 RELASI ANTAR ENTITI APP STORE DAN PLAY STORE	31
GAMBAR 4. 12 CARDINALITY RATIO DAN PARTICIPATION CONSTRAINT PADA APP STORE DAN PLAY STORE	32
GAMBAR 4.13 ATRIBUT-ATRIBUT YANG DIPERLUKAN DI APP DAN LIST PLAY STORE	32
GAMBAR 4.14 ATRIBUT-ATRIBUT YANG DIPERLUKAN DI HISTOGRAM PLAY STORE	33
GAMBAR 4.15 ATRIBUT-ATRIBUT YANG DIPERLUKAN DI KOMENTAR PLAY STORE	33
GAMBAR 4.16 ATRIBUT-ATRIBUT YANG DIPERLUKAN DI PERMISSION PLAY STORE	33
GAMBAR 4.17 ATRIBUT-ATRIBUT YANG DIPERLUKAN DI LAYAR TANGKAP PLAY STORE	33
GAMBAR 4.18 ATRIBUT-ATRIBUT YANG DIPERLUKAN DI REVIEW PLAY STORE	34
GAMBAR 4.19 ATRIBUT-ATRIBUT YANG DIPERLUKAN DI APP STORE	34
GAMBAR 4.20 PROSES GENERALISASI PADA PLAY STORE	35
GAMBAR 4.21 PROSES GENERALISASI PADA APP STORE	36
GAMBAR 4.22 MODEL DATA KONSEPTUAL	49
GAMBAR 4.23 ENTITAS <i>GOOGLEPLAY_SCRAP_LIST</i>	50
GAMBAR 4.24 MAPPING ERD <i>GOOGLEPLAY_SCRAP_LIST</i>	50
GAMBAR 4.25 ENTITAS <i>APPSTORE_SCRAP_LIST</i>	50
GAMBAR 4.26 MAPPING ERD <i>APPSTORE_SCRAP_LIST</i>	51
GAMBAR 4.27 ENTITAS <i>PLAYAPPLICATIONS</i>	51
GAMBAR 4.28 MAPPING ERD <i>PLAYAPPLICATIONS</i>	51
GAMBAR 4.29 ENTITAS <i>APPSTOREAPPLICATIONS</i>	52
GAMBAR 4.30 MAPPING ERD <i>APPSTOREAPPLICATIONS</i>	52
GAMBAR 4.31 WEAK ENTITY PADA <i>GOOGLEPLAY_SCRAP_APP</i>	53

GAMBAR 4.32 PENAMBAHAN APPID DI GOOGLEPLAY_SCRAP_APP	53
GAMBAR 4.33 WEAK ENTITY PADA GOOGLEPLAY_SCRAP_REVIEWS	54
GAMBAR 4.34 PENAMBAHAN APPID DI GOOGLEPLAY_SCRAP_REVIEWS	54
GAMBAR 4.35 WEAK ENTITY PADA GOOGLEPLAY_SCRAP_HISTOGRAMS	55
GAMBAR 4.36 PENAMBAHAN APPID DI GOOGLEPLAY_SCRAP_HISTOGRAMS	55
GAMBAR 4.37 WEAK ENTITY PADA GOOGLEPLAY_SCRAP_PERMISSIONS	56
GAMBAR 4.38 PENAMBAHAN APPID DI GOOGLEPLAY_SCRAP_PERMISSIONS	56
GAMBAR 4.39 WEAK ENTITY PADA GOOGLEPLAY_SCRAP_SCREENSHOTS	57
GAMBAR 4.40 PENAMBAHAN APPID DI GOOGLEPLAY_SCRAP_SCREENSHOTS ...	57
GAMBAR 4.41 WEAK ENTITY PADA GOOGLEPLAY_SCRAP_COMMENTS	58
GAMBAR 4.42 PENAMBAHAN APPID DI GOOGLEPLAY_SCRAP_COMMENTS	58
GAMBAR 4.43 WEAK ENTITY PADA APPSTORE_SCRAP_APP	59
GAMBAR 4.44 PENAMBAHAN APPID DI APPSTORE_SCRAP_APP	59
GAMBAR 4.45 RELASI 1:1 GOOGLEPLAY_SCRAP_APP DENGAN GOOGLEPLAY_SCRAP_LIST	60
GAMBAR 4.46 RELASI 1:1 APPSTORE_SCRAP_APP DENGAN APPSTORE_SCRAP_LIST	61
GAMBAR 4.47 PENAMBAHAN ENTITAS PLAYAPPSIMILARS	61
GAMBAR 4.48 HASIL AKHIR MODEL DATA LOGIKAL	62
GAMBAR 4.49 DESAIN BASIS DATA DI MYSQL	63
GAMBAR 4.50 ANTAR MUKA HALAMAN UTAMA APLIKASI	65
GAMBAR 4.51 ANTAR MUKA HALAMAN DETAIL APLIKASI	66
GAMBAR 4.52 ANTAR MUKA HALAMAN ULASAN APLIKASI	66
GAMBAR 4.53 ANTAR MUKA HALAMAN LOGIN	67
GAMBAR 4.54 ANTAR MUKA HALAMAN PENDAFTARAN AKUN	67
GAMBAR 4.55 ANTAR MUKA HALAMAN INSERT DATA APPID	68
GAMBAR 4.56 ANTAR MUKA HALAMAN INSERT DATA LENGKAP	68
GAMBAR 4.57 USE CASE	69
GAMBAR 5.1 STRUKTUR DIREKTORI APLIKASI	74
GAMBAR 5.2 HASIL AKHIR PROSES PEMASANGAN WEBSITE API	75
GAMBAR 5.3 KONFIGURASI MYSQL PDO	76
GAMBAR 5.4 KODE UNTUK MENGAMBIL DAFTAR KATEGORI	77
GAMBAR 5.5 KODE UNTUK MENDAPATKAN DAFTAR APLIKASI	77
GAMBAR 5.6 KODE UNTUK MENGONVERSI KARAKTER	78
GAMBAR 5.7 KODE UNTUK MEMASUKKAN DAFTAR APLIKASI KE DATABASE	78
GAMBAR 5.8 KODE UNTUK MENDAPATKAN APPID	79
GAMBAR 5.9 KODE UNTUK MEMECAH ARRAY DAN MENGONVERSI KARAKTER .	79
GAMBAR 5.10 KODE UNTUK MEMASUKKAN DATA KE BASIS DATA DENGAN UPDATE	80

GAMBAR 5.11 KODE UNTUK MENGAMBIL ID_AS DAN APPID DI APPSTORE	81
GAMBAR 5.12 KODE UNTUK MENGAMBIL DATA DI PLAYSTORE DENGAN KONDISI	81
GAMBAR 5.13 KODE UNTUK MENJALANKAN FUNGSI LEVENSHTAIN.....	82
GAMBAR 5.14 KODE UNTUK MEMASUKKAN DATA KE BASIS DATA DENGAN KONDISI	83
GAMBAR 5.15 KODE UNTUK MENAMPILKAN DAFTAR APLIKASI.....	84
GAMBAR 5.16 KUEI UNTUK MENAMPILKAN DAFTAR APLIKASI.....	85
GAMBAR 5.17 MODEL PLAYAPPSIMILAR	86
GAMBAR 5.18 KODE UNTUK MENDAPATKAN SCREENSHOTS APLIKASI	86
GAMBAR 5.19 KODE UNTUK MENAMPILKAN DETAIL APLIKASI	87
GAMBAR 5.20 KUEI UNTUK MENAMPILKAN DETAIL APLIKASI	87
GAMBAR 5.21 KUEI UNTUK MENAMPILKAN ULASAN APLIKASI	88
GAMBAR 5.22 MODEL REVIEWS	88
GAMBAR 5.23 KODE UNTUK MENAMPILKAN ULASAN APLIKASI.....	89
GAMBAR 5.24 KODE UNTUK INPUT FORM APPID.....	91
GAMBAR 5.25 KODE UNTUK MEMASUKKAN DETAIL APLIKASI DARI WEBSITE	91
GAMBAR 5.26 KODE UNTUK MENDAPATKAN DETAIL APLIKASI DARI INPUT FORM	91
GAMBAR 5.27 KODE UNTUK MENAMPILKAN TEXT FIELD DETAIL APLIKASI	92
GAMBAR 6.1 DATA DI TABEL PLAYAPPLICATIONS.....	93
GAMBAR 6.2 DATA DI TABEL COMMENTS.....	93
GAMBAR 6.3 DATA DI TABEL REVIEWS.....	94
GAMBAR 6.4 DATA DI TABEL HISTOGRAMS	94
GAMBAR 6.5 DATA DI TABEL SCREENSHOTS	94
GAMBAR 6.6 DATA DI TABEL PERMISSIONS.....	94
GAMBAR 6.7 DATA DARI TABEL APPSTOREAPPLICATIONS	95
GAMBAR 6.8 HALAMAN DAFTAR APLIKASI	96
GAMBAR 6.9 HALAMAN DETAIL APLIKASI.....	96
GAMBAR 6.10 HALAMAN ULASAN APLIKASI	97
GAMBAR 6.11 HALAMAN MASUKAN DATA SECARA RINCI	97
GAMBAR 6.12 HALAMAN MASUKAN DATA BERDASARKAN APPID.....	98
GAMBAR 6.13 HALAMAN LOGIN	98
GAMBAR 6.14 HALAMAN REGISTRASI.....	99

DAFTAR TABEL

TABEL 2.1 PAPER ACUAN 1	7
TABEL 2.2 PAPER ACUAN 2	8
TABEL 2.3 PSEUDOCODE LEVENSHTAIN DISTANCE	14
TABEL 4.1 DATA DICTIONARY PADA TABEL PLAYAPPLICATIONS.....	38
TABEL 4.2 DATA DICTIONARY PADA TABEL GOOGLEPLAY_SCRAP_COMMENTS...	41
TABEL 4.3 DATA DICTIONARY PADA TABEL GOOGLEPLAY_SCRAP_HISTOGRAMS .	41
TABEL 4.4 DATA DICTIONARY PADA TABEL GOOGLEPLAY_SCRAP_PERMISSIONS.	42
TABEL 4.5 DATA DICTIONARY PADA TABEL GOOGLEPLAY_SCRAP_REVIEWS.....	42
TABEL 4.6 DATA DICTIONARY PADA TABEL GOOGLEPLAY_SCRAP_SCREENSHOTS	43
TABEL 4.7 DATA DICTIONARY PADA TABEL APPSTOREAPPLICATIONS.....	44
TABEL 4.8 DAFTAR USE CASE	69
TABEL 4.9 <i>DESCRIPTIVE CASE</i> MEMILIH APLIKASI.....	70
TABEL 4.10 <i>DESCRIPTIVE CASE</i> MELIHAT DETAIL APLIKASI	70
TABEL 4. 11 <i>DESCRIPTIVE CASE</i> DAFTAR AKUN DEVELOPER	70
TABEL 4. 12 <i>DESCRIPTIVE CASE</i> LOGIN KE HALAMAN DEVELOPER	70
TABEL 4. 13 <i>DESCRIPTIVE CASE</i> INPUT DATA APLIKASI	71
TABEL 4. 14 <i>DESCRIPTIVE CASE</i> POSTING APLIKASI	71
TABEL 4. 15 <i>DESCRIPTIVE CASE UPDATE</i> DATA PLAY STORE.....	71
TABEL 4. 16 <i>DESCRIPTIVE CASE UPDATE</i> DATA APP STORE.....	71
TABEL 4. 17 <i>DESCRIPTIVE CASE</i> MEMASUKKAN DATA KE BASIS DATA	72
TABEL 5.1 SPESIFIKASI PERANGKAT KERAS	73
TABEL 5.2 SPESIFIKASI PERANGKAT LUNAK.....	73
TABEL 6. 1 DAFTAR TEST CASE.....	100
TABEL 6. 2 HASIL JUSTIFIKASI SYSTEM USABILITY SCALE	101

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

Dalam bab ini dijelaskan gambaran umum mengenai tugas akhir yang diangkat meliputi latar belakang masalah, perumusan masalah, batasan tugas akhir, tujuan tugas akhir dan relevansi atau manfaat kegiatan tugas akhir. Selain itu dijelaskan pula mengenai sistematika penulisan untuk memudahkan dalam membaca buku tugas akhir ini.

1.1 Latar belakang

Teknologi internet di masa modern ini semakin meningkat penggunaannya. Sampai saat ini, penggunaan internet mulai beralih ke arah mobile phone. Pada tahun 2016, jumlah pengguna smartphone di dunia tercatat hingga 2.1 miliar [1]. Hal ini menyebabkan akses ke layanan konten digital seperti *play store* dan *app store* meningkat. *Play store* dan *app store* adalah jenis *platform* distribusi digital untuk *mobile* dan aplikasi desktop. *App store* terus tumbuh dengan tingkat pertumbuhan yang tinggi dengan ribuan pengembang, ribuan aplikasi, dan keuntungan jutaan dollar yang didapatkan [2]. Google play sebagai *app store* terbesar telah memiliki aplikasi yang tersedia sebanyak 2.6 juta aplikasi di 2016 [3]. Google play store pada saat ini telah mencatat perhitungan kumulatif aplikasi yang sudah di unduh sebanyak 65 miliar kali dari Agustus 2010 hingga penghujung akhir Mei 2016 [4].

Seiring bertambahnya jumlah aplikasi yang melimpah pada layanan konten digital, proses pencarian aplikasi yang spesifik akan semakin sulit. Para pengguna aplikasi menginginkan cara yang efektif dan efisien dalam melakukan evaluasi berbagai aplikasi yang tersedia dengan cepat dan sesuai dengan kebutuhannya masing-masing. Namun saat ini, tidak jarang pengguna aplikasi yang mengalami kesulitan memilih layanan konten yang tepat dalam mengevaluasi aplikasi tertentu.

Sehingga pengguna memerlukan usaha lebih dalam menemukan informasi aplikasi yang ingin dievaluasi tersebut.

Selain itu, banyaknya aplikasi yang tersedia di berbagai layanan konten digital, tidak secara langsung memberikan dukungan bagi indie developer dalam mempublikasikan karyanya. Hal ini membuat para developer harus mampu mengembangkan produk-produknya dengan dana seadanya disertai publikasi sendiri. Publikasi mandiri yang dilakukan oleh para developer tentu tidak dapat dinikmati oleh berbagai kalangan masyarakat secara luas karena tidak terintegrasinya informasi aplikasi dalam suatu wadah yang dapat dinikmati banyak orang. Media ini dibutuhkan untuk memfasilitasi kedua belah pihak antara pengguna aplikasi dan *developer*.

Untuk menangani permasalahan tersebut, penelitian ini akan berfokus pada pengembangan sebuah prototype direktori software untuk melakukan penyajian katalog software kepada pengguna perangkat lunak dengan agregasi produk dari berbagai layanan konten digital seperti play store dan app store. *Software* direktori adalah media yang mengumpulkan berbagai konten perangkat lunak secara terintegrasi dari google play store dan app store. Metode yang dilakukan dalam prototype aplikasi ini yaitu *prototyping*. *Prototyping* adalah metodologi pengembangan perangkat lunak di mana model sistem dikembangkan dan dievaluasi semakin banyak digunakan untuk membantu menentukan persyaratan sistem dan kemampuan fungsional dengan cepat dan akurat. Metode *prototyping* digunakan karena fleksibilitas dalam pembangunan dari desain aplikasi yang akan dibangun.

Dengan adanya *prototype* ini diharapkan dapat mempermudah dalam mendapatkan informasi aplikasi multi platform secara detail dan lengkap serta dapat menjadi bahan pertimbangan dalam pembelian *device* di masa yang akan datang. Serta bagi developer dapat membantu mempublikasikan produknya dan dapat memasarkan produknya.

1.2 Rumusan masalah

Rumusan masalah dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Bagaimana cara yang efektif dan efisien dalam mendapatkan informasi aplikasi pada *app store*?
2. Bagaimana cara menangani kesulitan *indie developer* dalam mempublikasikan karyanya berupa aplikasi *multi platform*?

1.3 Batasan masalah

Sesuai dengan deskripsi permasalahan yang telah dijelaskan diatas, adapun batasan permasalahan dari penyelesaian tugas akhir ini adalah sebagai berikut:

1. Aplikasi merupakan berbasis Web yang dibangun dengan Laravel PHP Framework dan JavaScript.
2. Sumber data yang digunakan dalam aplikasi ini didapatkan hanya dari google play store dan app store masing-masing sejumlah 1000 produk.
3. Dalam tahap evaluasi menggunakan metode usability testing kepada 10 orang.

1.4 Tujuan

Tujuan utama dari pembuatan tugas akhir yang berfungsi sebagai berikut:

1. Sebagai direktori aplikasi yang berupa aplikasi web browser berbasis PHP atau HTML yang terstruktur dan memuat hasil ekstraksi dan agregasi produk.
2. Sebagai sarana publikasi aplikasi dari *developer* secara gratis.

1.5 Manfaat

Berikut manfaat yang diperoleh, dengan melihat dari dua belah sudut pandang, yaitu sudut pandang penulis dan pihak pengguna:

1.5.1 Bagi Konsumen

1. Mempermudah pencarian produk tanpa harus memiliki *device* yang bersangkutan sebagai bahan pertimbangan dalam memilih *device* di masa mendatang.
2. Mempermudah dalam mendapatkan informasi yang lebih lengkap dan detail dari suatu produk

1.5.2 Bagi *Developer* Produk Perangkat Lunak

Mempermudah *developer* dalam mengenalkan produknya, sehingga dapat membantu pemasaran produk.

1.6 Relevansi

Topik penelitian yang menjadi fokus dari tugas akhir ini adalah pengembangan sebuah *prototype* untuk melakukan penyajian *software* direktori kepada pengguna perangkat lunak dengan agregasi produk dari berbagai layanan konten digital seperti play store dan app store yang merupakan obyek penelitian dalam area Sistem Enterprise. Selain itu, terdapat beberapa mata kuliah yang terkait dengan penelitian tugas akhir ini sebagai berikut:

No.	Rumusan Masalah	Mata Kuliah
1.	Bagaimana pengguna dapat mengevaluasi produk yang efektif dan efisien??	Analisa dan desain perangkat lunak
2.	Bagaimana mengimplementasikan prototype aplikasi yang dapat menyajikan software direktori secara terstruktur?	Konstruksi dan Pengujian Perangkat Lunak, Integrasi Aplikasi Korporasi

(Halaman ini sengaja dikosongkan)

BAB II

TINJAUAN PUSTAKA

Pada bab ini berisikan tinjauan pustaka yang akan digunakan dalam penelitian tugas akhir ini, yang mencakup penelitian-penelitian sebelumnya, dasar teori dan metode yang digunakan selama pengerjaan.

2.1 Penelitian sebelumnya

Pada bagian ini akan dibahas penelitian-penelitian sebelumnya berupa jurnal, paten, maupun dokumen yang terkait dengan pengembangan penelitian tugas akhir ini, diantaranya:

Tabel 2.1 Paper Acuan 1

Dipublikasikan pada, Tahun	<i>Massachusetts Institute of Technology, 2005</i>
Judul	Information Aggregation using the Caméléon# Web Wrapper
Nama	Aykut Firat, Stuart Madnick, Nor Adnan Yahaya, Choo Wai Kuan, and Stéphane Bressan
Gambaran umum penelitian	Penelitian ini menggunakan tool agar proses ekstraksi dan agregasi menjadi efektif dan efisien. Ekstraksi dan agregasi menjadi landasan bagi banyak data yang ditaruh di dalam konteks dan berubah menjadi informasi Untuk menggambarkan fitur-fiturnya, digunakan contoh yang

	mengumpulkan agregasi biaya penerbangan data dari delapan situs online, Termasuk Travelocity, Orbitz, dan Expedia. penelitian ini mencakup integrasi Caméléon# dengan sistem komersial manajemen database, seperti MS SQL Server, dan bahasa XML query, seperti XQuery [1].
Keterkaitan penelitian	Penelitian ini memiliki hubungan dalam design dan implementasinya dimana proses ekstraksi dan agregasi data yang dilakukan setelah melakukan scraping yang dapat secara dinamis mengubah ke table relasi.

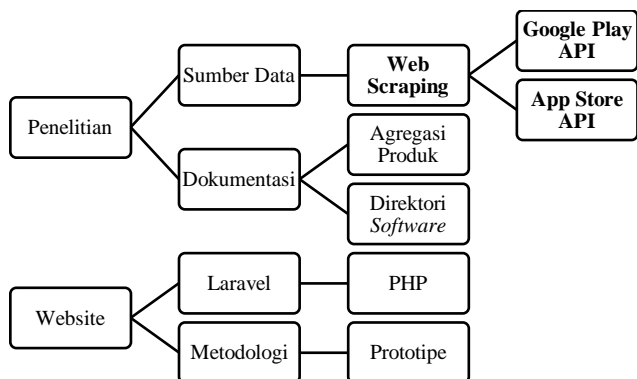
Tabel 2.2 Paper Acuan 2

Nomor Paten, Tahun	US7082426 B2, 2006
Judul	Content Aggregation Method and Apparatus For An On-line Product Catalog
Nama	Timothy Allen Musgrove, Robin Hiroko Walsh
Gambaran umum penelitian	Metode ini terdiri dari daftar informasi produk yang berasal dari banyak toko

	online yang menghubungkan id produk unik sesuai dengan produk yang terkait. Dalam hal ini dapat melakukan pembelian produk dalam satu halaman yang berasal dari macam-macam toko online [2].
Keterkaitan penelitian	Penelitian ini mengambil metode yang menghubungkan id produk unik dari berbagai macam toko online agar memiliki keterkaitan dalam detail produknya.

2.1 Landasan teori

Bagian ini membahas teori dan konsep yang berkaitan dengan penelitian tugas akhir. Relevansi dan dasar teori yang dibahas pada penelitian ini dapat dilihat pada Gambar 2. 1:



Gambar 2. 1 Bagan Dasar Teori

Landasan teori disusun untuk menjawab pertanyaan berikut:

1. Adakah cara cepat untuk mengambil data produk dari *Google play*?
2. Adakah cara cepat untuk mengambil data produk dari *App store*?
3. *Framework* apakah yang dapat digunakan untuk mempercepat pengembangan aplikasi web dan membantu merapikan arsitektur?
4. Bagaimana menguji rancangan yang dikembangkan kepada pengguna?
5. Bagaimana mencari kesamaan aplikasi di *Google play* dan *App store*?
6. Bagaimana melakukan pengambilan data yang dibutuhkan dalam membuat prototipe?

2.2.1 Google play store API

Google play store API adalah *software interface* yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk *library* dan menjelaskan bagaimana *Google play store* dapat berinteraksi dengan *software* lain.

Google play store API digunakan untuk memudahkan mendapatkan data dari *Google play store* dan juga untuk memudahkan pembangunan perangkat lunak yang berhubungan dengan *Google play store*. *Google play store API* menyediakan metode dimana metode pengambilan data tersebut dapat dilihat pada rincian berikut [3] :

1. *App*: Mengambil data aplikasi secara utuh.

```
GET: /app/:appId/:lang/:country
```

2. *List*: Mengambil daftar aplikasi dari suatu koleksi dalam Google Play.

```
GET: /list/:collection/:category/:age/:lang/:country/:num/:start
```

3. *Search*: Mengambil daftar aplikasi dari pencarian kata kunci.

```
GET: /search/:term/:lang/:country/:price/:num
```

4. *Developer*: Mendapatkan daftar aplikasi berdasarkan nama *developer*.

```
GET: /developer/:lang/:country/:num
```

5. *Suggest*: Mendapatkan 5 sugesti dari hasil pencarian kata kunci.

```
GET: /suggest/:term
```

6. *Reviews*: Mendapatkan data ulasan per aplikasi.

```
GET: /reviews/:appId/:lang/:sort/:page
```

7. *Similar*: Mendapatkan data aplikasi terkait dari kata kunci yang diinput.

```
GET: /similar/:appId/:lang
```

8. *Permissions*: Mendapatkan *permission* per aplikasi yang dispesifikan.

```
GET: /permission/:appId/:lang/:short
```

Hasil dari parameter dan metode ini adalah data Google Play yang ditampilkan dalam bentuk json yang sudah di *encode*.

2.2.2 App store API

App store API adalah *software interface* yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk *library* dan menjelaskan bagaimana *app store* dapat berinteraksi dengan *software* lain.

App store API digunakan untuk memudahkan mendapatkan data dari *app store* dan juga untuk memudahkan pembangunan perangkat lunak yang berhubungan dengan *App store*. *App store API* menyediakan metode dimana metode pengambilan data tersebut dapat dilihat pada rincian berikut [4] :

1. *App*: Mengambil data aplikasi secara utuh.

```
GET: /app/:id/:appID
```

2. *List*: Mengambil daftar aplikasi dari suatu koleksi dalam Google Play.

```
GET: /list/:collection/:category/:country/:num
```

3. *Search*: Mengambil daftar aplikasi dari pencarian kata kunci.

```
GET: /search/:term/:device/:num/:country
```

4. *Suggest*: Mendapatkan 5 sugesti dari hasil pencarian kata kunci.

```
GET: /suggest/:term
```

5. *Similar*: Mendapatkan data aplikasi terkait dari kata kunci yang diinput.

```
GET: /similar/:id/:appId/:country
```

6. *Reviews*: Mendapatkan data ulasan per aplikasi.

```
GET: /reviews/:id/:appId/:country/:page/:sort
```

2.2.3 Laravel

Laravel adalah sebuah Framework PHP 5.4 yang bersifat *opensource* yang ditulis oleh Taylor Otwell dengan lisensi dibawah MIT License [5]. Laravel dibuat untuk membantu para *developer* khususnya dalam membuat sebuah web dengan sintaks yang sederhana, elegan, ekspresif dan menyenangkan. Dengan Laravel, tugas-tugas umum *developer* dapat dikurangi pada sebagian besar proyek-proyek web seperti *routing*, *session* dan *caching*. Pada penelitian ini, tujuan utama dari laravel adalah mempermudah *coding* dalam membuat sebuah produk web. Ini membuktikan bahwa menggunakan Laravel memang dapat mempercepat dan mempermudah *development* website [6].

2.2.4 Usability testing

Usability testing adalah alat yang sangat berharga untuk mengevaluasi *interface* dari segi efektivitas dan mudah dioperasikan. Pengujian kebergunaan (*usability testing*) berdasarkan lima komponen yaitu *learnability*, *efficiency*, *memorability*, *errors* dan *satisfaction*[7]. Pada penelitian ini, Tahap-tahap yang dilakukan untuk melakukan *usability testing* diantaranya menentukan evaluator, melakukan survei dengan kuisioner, melakukan pemilihan fungsi, membuat tugas *usability test*, menyusun skrip *usability test*, melaksanakan *usability test* dengan wawancara, melakukan analisa data dari hasil *usability test*, dan menyusun rekomendasi perbaikan aplikasi [8].

2.2.5 Levenshtein Distance

Levenshtein Distance (LD) adalah sebuah ukuran dari kemiripan atau kecocokan antar dua String, di mana kita akan mengacu sebagai sumber atau disebut String (s) dan target atau disebut String t [9]. Distance yang dimaksud adalah jumlah dari penghapusan (deletion), penambahan (insertions) atau penggantian (substitution) yang dibutuhkan untuk merubah String s menjadi String t [9]. Contohnya adalah sebagai berikut:

1. Jika s adalah "test" dan t adalah "test", maka $LD(s,t) = 0$, karena tidak ada transformasi yang dibutuhkan, karena $s = t$.
2. Jika s adalah "test" dan t adalah "tent", maka $LD(s,t) = 1$, karena dibutuhkan 1 substitusi untuk membuat $s = t$, sehingga cost dari Levenshtein adalah 1.

Secara umum, algoritma dari Levenshtein Distance sendiri berdasarkan [9] dapat dijelaskan dari Tabel 2.3 berikut:

Tabel 2.3 Pseudocode Levenshtein Distance

Step	Description
1	Set n to be the length of s. Set m to be the length of t. If $n = 0$, return m and exit. If $m = 0$, return n and exit.

	Construct a matrix containing 0..m rows and 0..n columns.
2	Initialize the first row to 0..n. Initialize the first column to 0..m.
3	Examine each character of s (i from 1 to n).
4	Examine each character of t (j from 1 to m).
5	If $s[i]$ equals $t[j]$, the cost is 0. If $s[i]$ doesn't equal $t[j]$, the cost is 1.
6	Set cell $d[i,j]$ of the matrix equal to the minimum of: a. The cell immediately above plus 1: $d[i-1,j] + 1$. b. The cell immediately to the left plus 1: $d[i,j-1] + 1$. c. The cell diagonally above and to the left plus the cost: $d[i-1,j-1] + \text{cost}$.
7	After the iteration steps (3, 4, 5, 6) are complete, the distance is found in cell $d[n,m]$.

2.2.6 Web Scraping

Web scraping adalah mengambil konten website dari sebuah situs secara otomatis tanpa harus menyalinnya secara manual untuk kemudian dipilah-pilah sesuai dengan informasi yang dibutuhkan. Cara yang bisa digunakan adalah dengan meng-copy paste sebuah website yang dilakukan seseorang secara manual, atau dengan cara otomatis yaitu memarsing seluruh isi website dengan tools atau aplikasi [10]. Pada penelitian ini, tujuan dari *web scraping* adalah untuk mencari informasi tertentu dan kemudian mengumpulkannya kedalam format yang diinginkan.

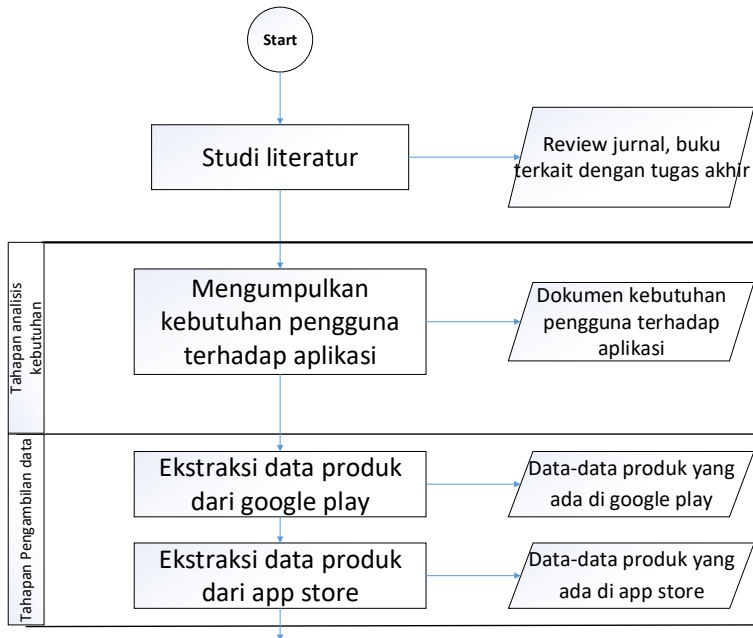
(Halaman ini sengaja dikosongkan)

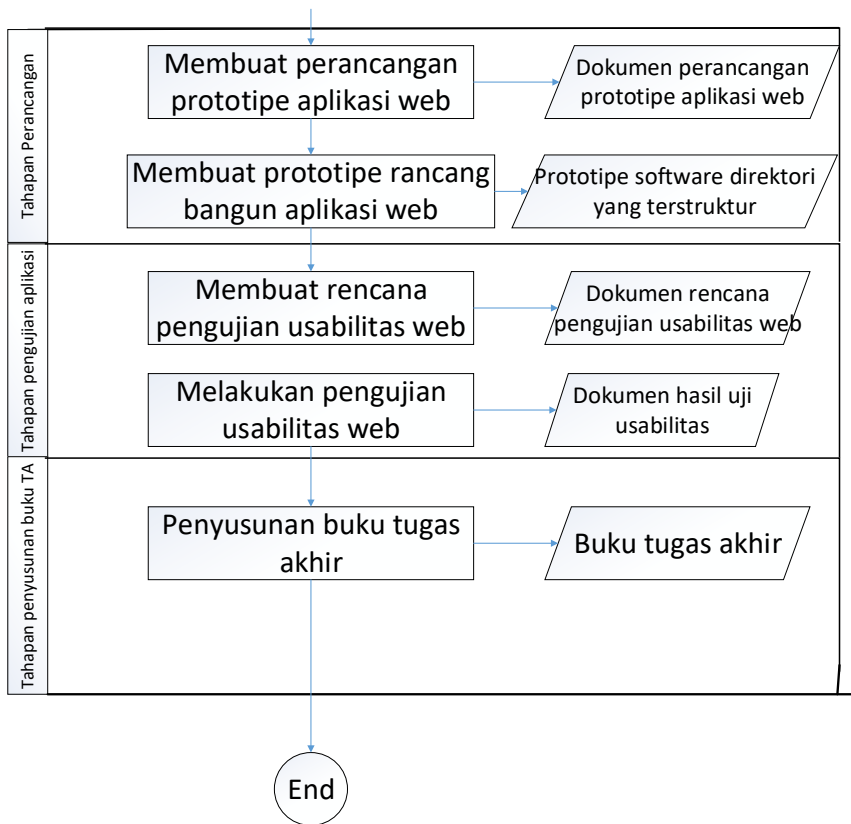
BAB III

METODOLOGI PENELITIAN

Pada bab ini dijelaskan mengenai gambaran metode dan alur pengerjaan tugas akhir. Penelitian ini menggunakan metode *Prototyping* untuk pengkajian, perancangan dan pengujian. Berdasarkan penelitian sebelumnya, metode prototyping sesuai dengan pola pengerjaan tugas akhir ini karena fleksibilitas dalam pembangunan dari desain aplikasi yang akan dibangun. Adapun tahapan dari penelitian ini sebagai berikut:

3.1 Diagram Metodologi





Gambar 3.1 Metodologi Penelitian

3.2 Uraian Metodologi

Pada bagian ini dijelaskan tahapan-tahapan pembuatan tugas akhir secara rinci sesuai dengan metodologi yang telah digambarkan sebelumnya.

3.2.1 Studi Literatur

Studi literatur dilakukan untuk mendapatkan informasi, wawasan, dan pengetahuan yang berguna dalam pembuatan tugas akhir. Studi literatur yang dilakukan meliputi pengetahuan dalam pengambilan data *google play* dan *app store* dengan *web*

scraping, pengetahuan terkait pemrosesan data dengan ekstraksi, agregasi dan penyajian data. Studi literatur yang dilakukan pada penelitian ini bersumber dari berbagai macam media yaitu jurnal internasional, tugas akhir, *website*, serta dokumen lainnya terkait studi kasus.

3.2.2 Analisis Kebutuhan

Analisis kebutuhan dilakukan dengan cara melakukan kuesioner kepada 30 orang yang memiliki preferensi untuk mengevaluasi perangkat lunak yaitu orang yang memiliki kedua konten digital baik *app store* dan juga *google play* sehingga pernah memiliki pengalaman dalam mengevaluasi perangkat lunak yang ada pada kedua layanan konten digital tersebut. Selain itu, wawancara dilakukan kepada 2 orang *indie developer* untuk mengetahui kebutuhannya terhadap *system* terkait publikasi karya berupa perangkat lunak dalam suatu media. Hasil wawancara nantinya akan digunakan sebagai bahan untuk menentukan kebutuhan fungsional sistem yang akan dirancang agar sesuai dengan kebutuhan pengguna.

3.2.3 Pengambilan Data

Setelah melakukan analisis kebutuhan, lalu dilakukan pengambilan data yang dilakukan dengan cara *scraping*. Data yang diambil berasal dari *google play* dan *app store*. Data yang didapatkan meliputi nama aplikasi dan detail deskripsi aplikasi yang ada dalam layanan konten digital. Pengumpulan data dilakukan dengan memanfaatkan *Application Programming Interface (API)* yang ada berasal dari penelitian yang dilakukan oleh pihak laboratorium *system enterprise*. Data yang telah dikumpulkan disimpan dalam database system.

3.2.4 Perancangan Prototipe

Setelah menyelesaikan pengambilan data, maka tahap selanjutnya adalah pembuatan prototipe *website* agregasi produk dengan fungsi yang spesifik terhadap direktori aplikasi (produk). Tujuan utama dari proses ini adalah untuk memberikan pengalaman terbaik dalam melakukan evaluasi

produk yang mudah dan efisien melalui pembuatan aplikasi web berbasis PHP/HTML yang menyajikan software direktori secara terstruktur. Prototipe dibuat dengan membuat draft antar muka pengguna yang merepresentasikan berbagai fungsi yang telah direncanakan pada tahap selanjutnya. Luaran dari tahapan perancangan prototipe adalah mockup web aplikasi berbasis PHP/HTML yang mampu menyajikan software direktori yang terstruktur kepada pengguna perangkat lunak.

Langkah selanjutnya dilakukan analisis proses bisnis dengan cara membuat desain UML diagram yaitu use case diagram dan class diagram.

3.2.5 Pengujian Aplikasi

Pada tahapan ini aplikasi akan diuji usability desain, luaran, dan fiturnya. Kategori pengujian yang dilakukan yaitu pengujian usability menggunakan *SUS matrix*. Pengujian *usability* dilakukan kepada 10 orang responden. Parameter pengujian dilakukan dengan membuat parameter ketercapaian. Apabila dari hasil pengujian didapatkan nilai dibawah dari ketentuan maka desain dan fitur diubah sesuai dengan harapan pengguna, yang dalam konteks tugas akhir ini akan menjadi bahan evaluasi di masa yang akan datang.

3.2.6 Penyusunan Buku Tugas Akhir

Penyusunan buku tugas akhir akan menghasilkan buku tugas akhir. Penyusunan buku ini dilakukan bersamaan dengan tahapan penelitian lainnya. Harapannya, buku tugas akhir ini dapat dijadikan referensi bagi penelitian selanjutnya.

3.3 Jadwal Kegiatan

Dalam proses pengerjaan tugas akhir ini terdapat jadwal pengerjaan. Berikut ini merupakan tabel rincian jadwal kegiatan selama penelitian tugas akhir berlangsung.

(Halaman ini sengaja dikosongkan)

BAB IV

PERANCANGAN

Pada bab ini dijelaskan perancangan awal yang diperlukan sebelum melakukan penelitian tugas akhir. Bab ini mencakup metode-metode dalam persiapan penelitian tugas akhir disertai penjelasannya

4.1 Pengambilan Data

Bagian ini akan menjelaskan pengambilan data sebagai acuan pada saat akan melakukan implementasi pada penelitian ini sehingga terjadinya kesalahan pada proses implementasi dapat diminimalisir.

4.1.1 Perancangan Pengambilan Data

Pada bagian ini akan dijelaskan mengenai perancangan pengambilan data yang bertujuan dalam pengolahan data mentah yang nantinya akan diimplementasikan pada penelitian ini.

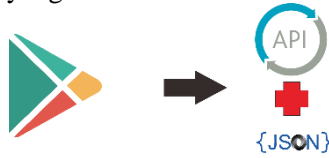
4.1.1.1 Identifikasi Karakteristik Situs

Situs yang ingin dilakukan pengambilan data dari *play store* dan *app store* yaitu *website API* yang di *deploy* ke *pstoreapi.herokuapp.com* dan *astoreapi.herokuapp.com*. *Website API* adalah situs yang datanya berasal dari *play store* dan *app store* dan menjadi jembatan bagi pengguna yang ingin mengambil data aplikasi. *Website API* digunakan demi kemudahan pengambilan data, mempercepat proses pengambilan, dan menghemat *bandwidth* saat pengambilan data. Konten di dalam *website* ini yaitu *JSON text* yang berisi kumpulan *array*. *Website API* pada *play store* membagi datanya menjadi *list* aplikasi, *detail* aplikasi, *review*, dan *permission*. Namun pada bagian *website API app store* hanya terdapat *list* aplikasi dan *detail* aplikasi karena kegagalan menjembatani ke *app store*.

4.1.1.2 Langkah Utama Memasukkan Data ke Sistem

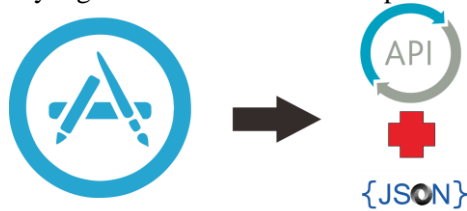
Pengambilan data dilakukan untuk mencari persamaan aplikasi yang ada di portal konten digital seperti *Play Store* dan *App Store*. Data yang diambil dari portal konten digital itu mengenai detail aplikasi yang ditampilkan di situs tersebut. Data yang diambil tidak langsung ke situs *play store*, melainkan dari *API* yang menggunakan *JSON*. *API* ini

berasal dari `gooplayapi.herokuapp.com` yang dilakukan oleh Natanael Yabes W[11]. Proses yang ada tersebut bisa dilihat pada Gambar 4.1:



Gambar 4.1 Alur Pengambilan Data Sebelum dimodifikasi di Play Store

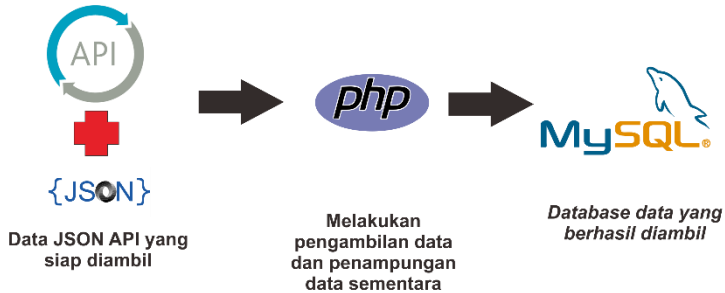
Pada penelitian ini, API yang menuju ke *app store* juga dibuat oleh penulis dan program API itu sendiri dimasukkan ke situs `applestoreapi.herokuapp.com` oleh Natanael Yabes W. untuk dibandingkan. Proses yang ada tersebut bisa dilihat pada Gambar 4.2.



Gambar 4.2 Alur Pengambilan Data Sebelum dimodifikasi di App Store

Konsep API dalam pengambilannya dilakukan secara *real time* ke situs portal konten digital. Pengambilan data dilakukan melalui curl client yang menggunakan json yang sudah disediakan di *google play api* dan *app store api*, lalu dimasukkan masing-masing ke basis data *google play API* dan *app store API*.

Data yang dimasukkan ke basis data ini bersifat umum dan spesifik. Data umum ini adalah data dengan banyak aplikasi namun hanya menampilkan data secara umum mengenai aplikasi tersebut. Data spesifik adalah data yang berisikan satu aplikasi namun menampilkan data secara rinci. Data general dan spesifik ini dijadikan 1 tabel *application* pada basis data dengan atribut yang dapat dilihat pada tabel yang memiliki alur proses pengambilan data dari API JSON yang dapat dilihat pada Gambar 4.3.



Gambar 4.3 Alur Pengambilan Data dari API JSON

Untuk lebih jelasnya berikut strategi memasukkan data ke sistem:

1. Memastikan `pstoreapi.herokuapp.com` dan `astoreapi.herokuapp.com` tidak mengalami pemblokiran konten sementara atau tidak terjadi *down server*.
2. Melakukan konfigurasi pada `php.ini` dengan mengubah `max_execution_time` menjadi 604800 karena akan dijalankan selama 1 minggu.
3. Mengubah `upload_max_filesize` dan `post_max_size` menjadi 500M dikarenakan pada proses pengambilan data menjalankan di banyak komputer demi mempercepat pengerjaan dan menghindari error filesize saat proses penyatuan.
4. Mengubah `memory_limit` pada `php.ini` menjadi 2048M dikarenakan pada proses pengambilan data membutuhkan banyak memori untuk menampung data sementara sebelum dimasukkan ke basis data.
5. Membuat tabel kategori pada basis data yang berisi 58 kategori demi memudahkan pengambilan data pada 58 kategori menggunakan perulangan.
6. Pengambilan data keseluruhan yang ada di *app*, *screenshots*, *comments*, *reviews*, *permissions*, dan *histograms* membutuhkan `appId` sehingga pengambilan data *list* dijalankan terlebih dahulu saat mengambil data.

```
[
- {
  url: "https://play.google.com/store/apps/details?id=com.facebook.orca",
  appId: "com.facebook.orca",
  title: "Messenger",
  summary: "Messenger – a faster way to message.",
  developer: "Facebook",
  icon: "//lh5.ggpht.com/0VYAvZLR9YhosF-thqm8x18EhsCfrEY_uk2og2F59K8IOx5TFpsXjFVwxaHvNUbEjc=w340",
  score: 4,
  price: "0",
  free: true
},

```

Gambar 4.4 Sampel data *list* Play Store API

Gambar 4.4 menunjukkan sampel data *list* yang datanya akan diambil, kemudian hasil tersebut diletakkan pada penyimpanan sementara di PHP. Lalu data yang tersimpan tadi kemudian di masukkan ke MYSQL, dari data yang tersimpan terdapat sebanyak 44100 aplikasi *play store* yang dapat dikumpulkan.

```
- histogram: {
  1: 263934,
  2: 100772,
  3: 209745,
  4: 560817,
  5: 3057461
},
offersIAP: true,
adSupported: false,
androidVersionText: "Varies with device",
androidVersion: "VARY",
contentRating: "Teen",
- screenshots: [
  "//lh3.googleusercontent.com/qJEqftz-pCdJMEaJ95ezNN-NAX9tEbsM3xaklyTmr8MkvQYQD_Ps6Van8AHraRwCdH-h310",
  "//lh3.googleusercontent.com/j9gGmD14KCK167W7xZqTVZ0sEOg2JU0dCpZ6AhsIIQLvqsPrqUxpKMjPn1a29pNUw-h310",
  "//lh3.googleusercontent.com/_ydtXHp3aSP0nQ25H5BwtGhXyaIX30V1F4sSETYPzm20EXIF40aNCtTijjycO3h51xA-h310",
  "//lh3.googleusercontent.com/38_8yP5MIiipxyKo5Xm0IwGeAlGx0b1ZSglaUava30T1kqEcPEoY5U5LSMC387E0Q-h310",
  "//lh3.googleusercontent.com/FlqCWUlltaeyrWJR0LXYrWxbC_TlCtPffjePlb3p4ol0hy83iZyO4-w_F5oGoqlMom0-h310",
  "//lh3.googleusercontent.com/Uz95MjjraP_DyGIUKHMMDVgXqY8WYVWYht0mu5WaatZ8yen1n3z7RxiUE0de67mhI-h310",
  "//lh3.googleusercontent.com/3q3jMwlp35WC_OXeRqmm03THIqLYmxzqP4kKKHdedtBb0G0FcUnIRkdhMdznsV35Ly-h310",
  "//lh3.googleusercontent.com/Q5r4Lrb1Z0rGr_7nh_VSP9Nc_Mb_alXakJ_ZvvgngS6j9Q7rswMnBaCe2b3TzSlupk-h310",
  "//lh3.googleusercontent.com/OpdJL984b_EV26RoDInPy2ZreDwXvL9j8C_eUVBNbiqT1k1BbP1yCUiCsCK0byRYW-bc-h310",
  "//lh3.googleusercontent.com/x_b71BM9RD8pkDMx62677Z2fg2UQj6Mpg0Qxx65wPqKAEMDLZEO_cWZHFwA51vRgA-h310",
  "//lh3.googleusercontent.com/OBwP0wW5Fe6qRaHOIYENNgWwvncBkvenxdErDeoRpGrg4MKvHKiV0A8I_8RubSNFXiIm-h310",
  "//lh3.googleusercontent.com/b6EfyKTm-jFrXB3nGzYqPuZx8Y_MXSvVWhFY1WzuP51n-FvG3vcrRtrIjyMD2eyS3g-h310",
  "//lh3.googleusercontent.com/-eqIow054ZS2LoADaMgh55h3Cv-oJhFbnc_RoaPi0Eqi1f15h5mC0Srwdyscx1d9AU-h310",
  "//lh3.googleusercontent.com/tz38PpsFjn3Yc0y4zHFuu17cQ4vXxx11lxv6YPF63092TImDFQ5V8WAg_g9v1Q5y4s-h310",
  "//lh3.googleusercontent.com/nDD_OXWf_I1NIxva51962w1zQxmN57Q63STXe-F4AVXsbz7sf1R109CXJ4k1GmpPrH-h310"
],
- comments: [
  "Please allow downloads. \"Paradox of choice\", is a very poor excuse for not allowing media downloads on Ne

```

Gambar 4.5 Sampel data *app* Play Store API

Gambar 4.5 menunjukkan sampel data *app*, *comments*, *histogram*, *screenshots* yang tersimpan jumlahnya menyamai *list* yaitu 44100 data aplikasi.

```
- {
  id: "gp:AQqpTOFzQFDSMqbvD3zr5rNMnSuci8IQ6Ig5b0e_xeKgqrN-zIVQoE-Ymz420531_lMhm3ASR93AmJZvL6g8Q",
  userName: "Annie Gill",
  userImage: "https://lh6.googleusercontent.com/-hX5spkdR47W/AAAAAAAAAT/AAAAAAAAAA/tWmvaYPOgio/v96-h96-p/photo.jpg",
  date: "July 2, 2017",
  url: "https://play.google.com/store/apps/details?id=com.spotify.music&reviewId=73AG0U9xcFRPNpRRkRTTXFidkRkenIck5NbJv1YzF10E1RNklnU2IuZV94ZUtncXJ0LXp2V1FvRS17bXo0MJ8TM",
  score: 3,
  title: "",
  text: "My downloads keep disappearing from my library for no reason, extremely frustrating."
},
```

Gambar 4.6 Sampel data *review* Play Store API

Gambar 4.6 menunjukkan sampel data *review* yang akan diambil, data *review* yang tersimpan sebanyak 473544 ulasan dari 44100 aplikasi *play store*.

```
- {
  permission: "add or remove accounts",
  description: "Allows the app to perform operations like adding and
},
- {
  permission: "read your own contact card",
  description: "Allows the app to read personal profile information s
identify you and may send your profile information to others."
},
- {
  permission: "read your contacts",
  description: "Allows the app to read data about your contacts store
in other ways with specific individuals. This permission allows app
knowledge."
},
- {
  permission: "modify your contacts",
  description: "Allows the app to modify the data about your contacts
communicated in other ways with specific contacts. This permission
},
```

Gambar 4.7 Sampel data *permission* Play Store API

Gambar 4.7 menunjukkan sampel data *permission* yang akan diambil, data *permission* yang tersimpan sebanyak 680927 perizinan dari 44100 aplikasi *play store*.

Gambar 4.9 menunjukkan data lengkap aplikasi yang berisi hanya 1 aplikasi saja per halaman di *app store* API.

4.1.2 Desain Basis Data

Pada bagian ini akan dijelaskan mengenai desain basis data yang memiliki tujuan dalam merancang basis data yang terstruktur dengan membuat relasi antar tabel yang pada pengerjaannya memiliki berbagai tahapan agar kolom-kolom dalam tabel tersebut hanya berisikan data-data yang tidak redundan.

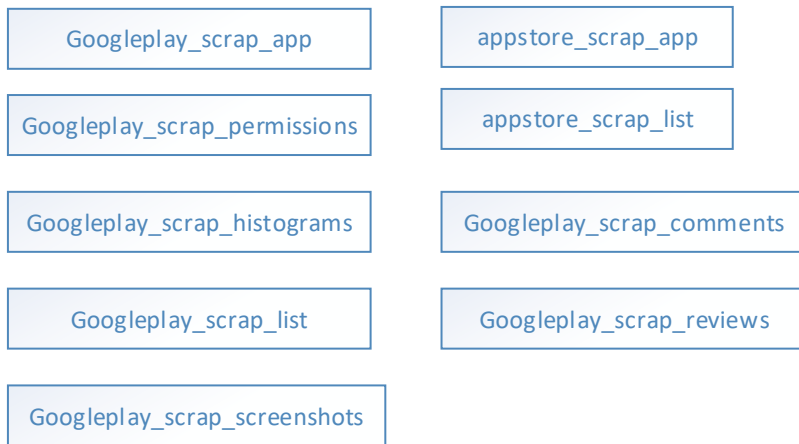
4.1.2.1 Perancangan Basis Data Konseptual

Perancangan basis data konseptual [12] adalah proses membuat suatu model informasi yang digunakan pada suatu perusahaan, yang terlepas dari semua pertimbangan fisik. Langkah-langkah didalam perancangan basis data konseptual, yaitu Membangun model data konseptual untuk setiap *user view*.

4.1.2.1.1 Identifikasi Tipe Entitas

Pada tahapan ini bertujuan untuk mengidentifikasi tipe utama yang diperlukan dari sudut pandang tertentu. Salah satu cara yang dapat digunakan dengan mengidentifikasi kata benda utama dan objek-objek yang berkaitan dengan spesifikasi.

Pada Gambar 4.10 Entitas *googleplay_scrap_list* dan *appstore_scrap_list* berisikan daftar aplikasi dari koleksi yang ada di dalam toko *online* tersebut, sehingga kedua entitas ini memiliki sifat umum. Selanjutnya, entitas *googleplay_scrap_app* dan *appstore_scrap_app* berisikan data aplikasi secara utuh dan menyeluruh, sehingga kedua entitas ini bersifat spesifik.

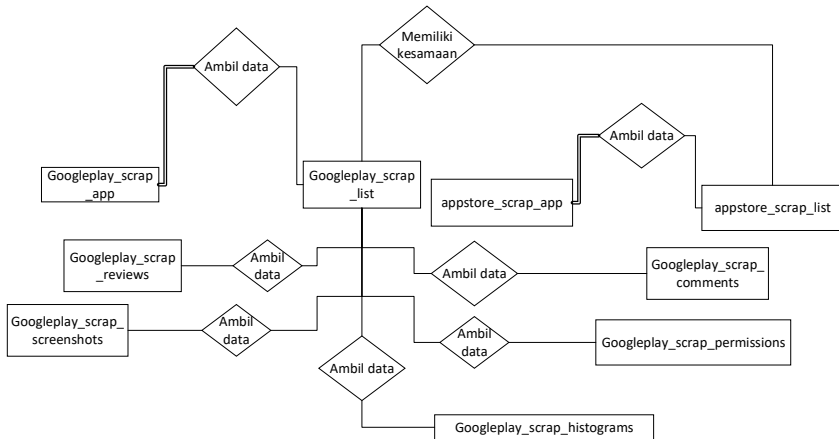


Gambar 4.10 Entitas app store dan play store

4.1.2.1.2 Identifikasi tipe relasi

Pada tahapan ini bertujuan untuk mengidentifikasi relasi penting yang ada antar berbagai tipe entiti yang telah diidentifikasi. Salah satu metode yang digunakan adalah dengan mempelajari kata kerja yang berkaitan dengan entiti, tipe relasi, dan spesifikasi.

Gambar 4.11 berikut ini adalah relasi antara entitas googleplay_scrap list dan googleplay_scrap_app, relasi antara entitas appstore_scrap_list dan appstore_scrap_app, relasi antara entitas googleplay_scrap list dan googleplay_scrap_app, relasi antara entitas googleplay_scrap list dan googleplay_scrap_comments, relasi antara entitas googleplay_scrap list dan googleplay_scrap_histograms, relasi antara entitas googleplay_scrap list dan googleplay_scrap_permissions, relasi antara entitas googleplay_scrap list dan googleplay_scrap_reviews, relasi antara entitas googleplay_scrap list dan googleplay_scrap_screenshots, dan juga googleplay_scrap_list dan appstore_scrap_list.



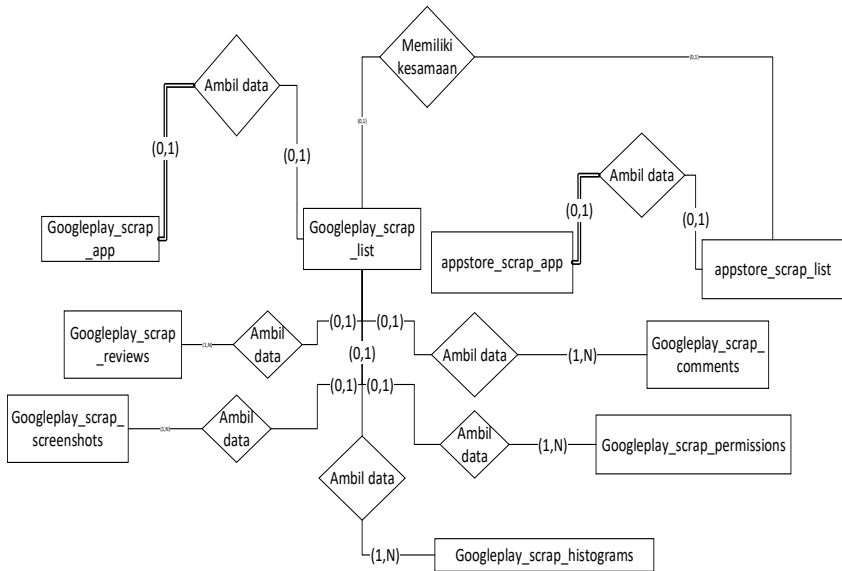
Gambar 4.11 Relasi antar entiti app store dan play store

4.1.2.1.3 Identifikasi dan menghubungkan atribut dengan entiti ataupun dengan tipe relasi

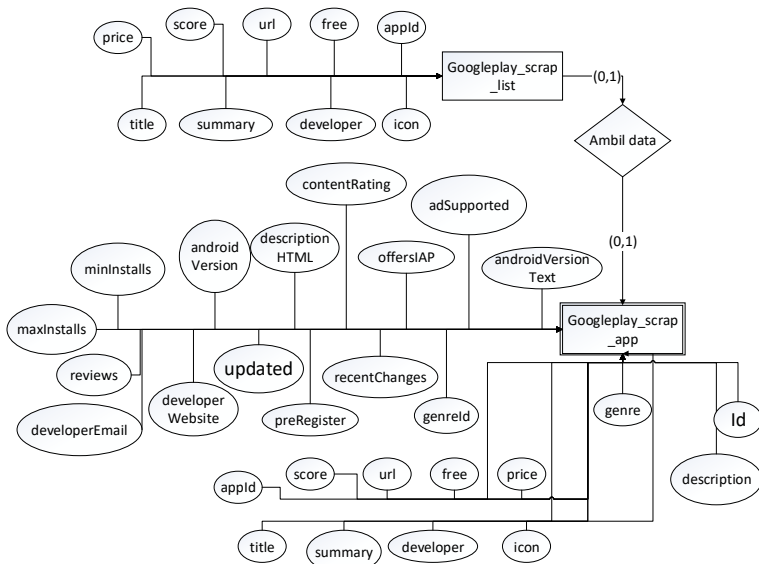
Pada tahapan ini bertujuan untuk menghubungkan atribut dengan tipe entitas dan relasi yang sesuai yang ditunjukkan pada Gambar 4. 12. Dalam menentukan atribut harus mampu mengidentifikasi *simple* atau *composite* atribut, *single* atau *multi-valued* atribut dan atribut turunan.

4.1.2.1.4 Menentukan domain atribut

Pada tahapan ini bertujuan untuk menentukan wewenang bagi atribut dalam model data konseptual. Hal ini ditunjukkan pada Gambar 4.13 sampai Gambar 4.19 dimana terdapat penambahan atribut dari masing-masing entitas. Domain adalah satu kumpulan nilai dimana menggambarkan nilai-nilai dari satu atau lebih atribut.



Gambar 4. 12 cardinality ratio dan participation constraint pada app store dan play store



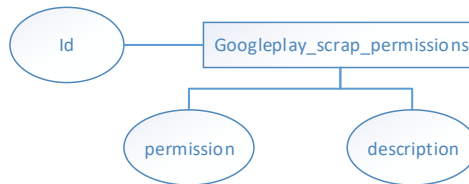
Gambar 4.13 atribut-atribut yang diperlukan di app dan list play store



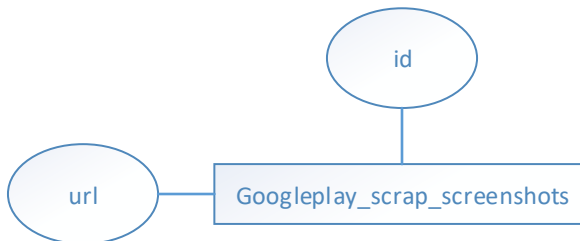
Gambar 4.14 atribut-atribut yang diperlukan di histogram play store



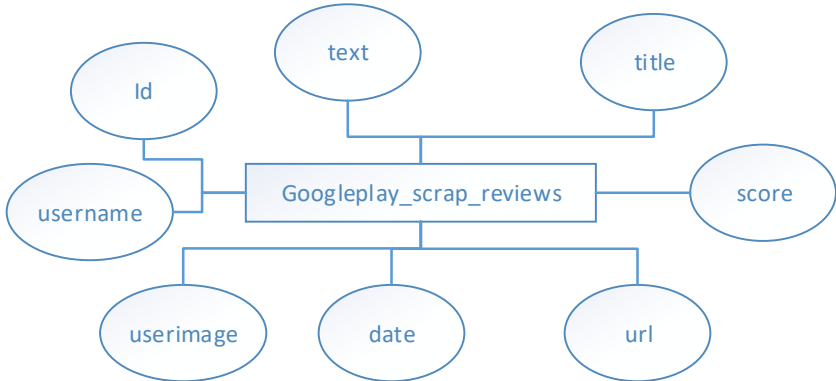
Gambar 4.15 atribut-atribut yang diperlukan di komentar play store



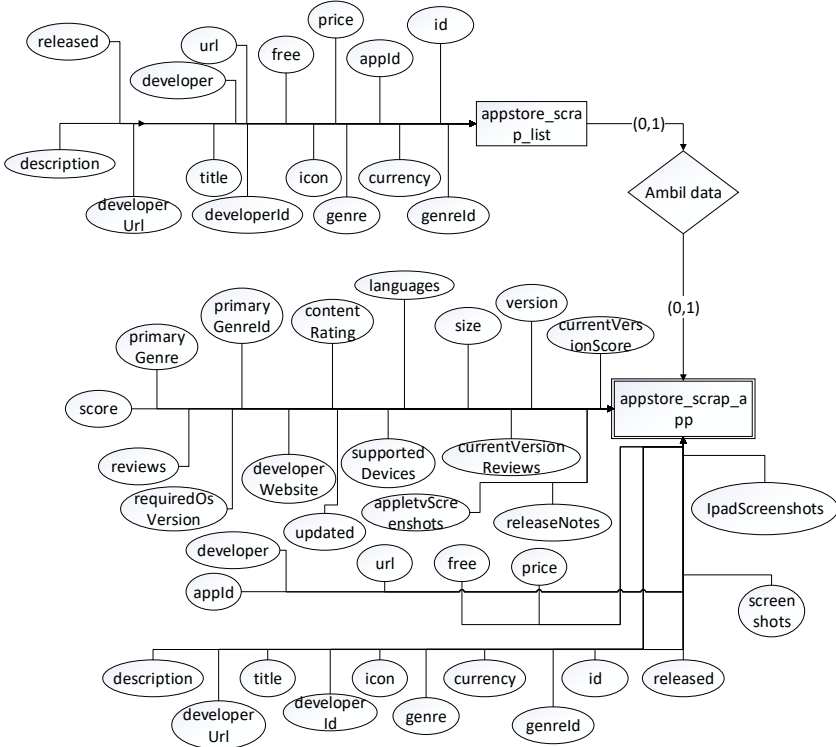
Gambar 4.16 atribut-atribut yang diperlukan di permission play store



Gambar 4.17 atribut-atribut yang diperlukan di layar tangkap play store



Gambar 4.18 atribut-atribut yang diperlukan di review play store



Gambar 4.19 atribut-atribut yang diperlukan di app store

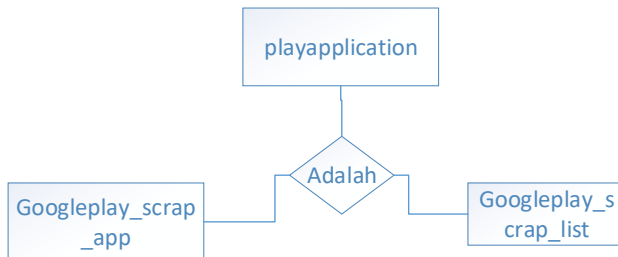
4.1.2.1.5 Menentukan atribut candidate key dan primary key

Pada bagian ini bertujuan untuk mengidentifikasi *candidate key* untuk setiap tipe relasi, dan jika terdapat lebih dari satu *candidate key*, maka dipilih salah satu untuk menjadi *primary key*.

Dalam hal ini, untuk googleplay_scrap_app memiliki *primary key* yaitu id dan *candidate key* yaitu appId. Lalu appstore_scrap_app memiliki *primary key* yaitu id dan *candidate key* yaitu appId.

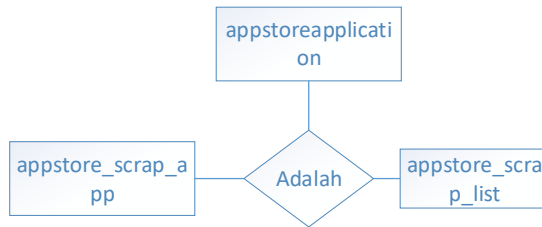
4.1.2.1.6 Mempertimbangkan penggunaan konsep pemodelan yang lebih baik

Pada tahapan ini bertujuan untuk mempertimbangkan konsep pemodelan yang lebih baik seperti spesialisasi, generalisasi, agregasi, atau komposisi. Namun pada tahapan ini, yang terjadi hanyalah proses generalisasi, dimana terjadi proses peleburan di tabel googleplay_scrap_app dan googleplay_scrap_list menjadi tabel playapplication. Ini dikarenakan atribut yang ada di googleplay_scrap_list memiliki kesamaan terhadap googleplay_scrap_app yang bisa dilihat pada Gambar 4.20. Demi menghemat memori dan menghilangkan redundansi, maka dibutuhkan proses generalisasi.



Gambar 4.20 Proses Generalisasi pada Play Store

Demikian juga pada tabel-tabel di app store, pada Gambar 4.21 dilakukan proses generalisasi pada appstore_scrap_app dan appstore_scrap_list menjadi appstoreapplication.



Gambar 4.21 Proses Generalisasi pada App Store

4.1.2.1.7 Memeriksa dengan model redundansi

Pada tahapan ini bertujuan untuk memeriksa keberadaan redundansi dalam model basis data. Aktivitas yang dilakukan pada tahapan ini adalah menentukan ulang relasi 1:1 dan membuang relasi yang redundan. Namun pada tahapan ini, tidak ada perubahan pada basis data konseptual dikarenakan kondisi tabel yang ada tidak memerlukan perlakuan dua aktivitas pada tahapan ini.

4.1.2.1.8 Data Dictionary

Data dictionary bisa juga disebut *system catalog* dan bisa diartikan menjadi hasil kompilasi *data definition language* yang kumpulan tabelnya tersimpan secara kolektif.

Biasanya, di dalam sebuah *data dictionary* terdapat beberapa hal berikut:

1. Definisi dari semua objek skema yang terdapat di dalam sebuah database (tabel, view, indeks, cluster, sinonim, urutan, prosedur, dll)
2. Berapa banyak alokasi data yang digunakan oleh objek skema tersebut
3. Nilai default untuk setiap kolom
4. Nama dari pengguna
5. Keuntungan dan peran dari setiap pengguna yang telah diperbolehkan
6. Informasi yang telah diaudit, seperti siapa yang dapat mengakses atau memperbaharui objek skema tersebut
7. Dan informasi umum mengenai basis data lainnya

Data dictionary biasanya mempunyai tiga pengguna utama, yaitu :

1. Administrator yang dapat mengakses *data dictionary* tersebut untuk mengetahui informasi mengenai pengguna, objek skema dan struktur penyimpanan dari basis data
2. Administrator yang dapat memodifikasi *data dictionary* setiap kali ada DDL (*Data Definition Language*) yang ditambahkan
3. Pengguna lainnya yang dapat menggunakan *data dictionary* sebagai referensi yang dapat dibaca untuk mengetahui informasi apapun mengenai basis data

Pembuatan *data dictionary* berguna untuk mengetahui jenis tipe apa yang akan digunakan pada setiap kolom data pada basis data yang digunakan dan penjelasan lebih rinci dari ERD (*Entity Relation Diagram*) dari sebuah basis data. Pada penelitian kali ini, akan ada dua tabel yang akan digunakan yaitu tabel *playapplication* untuk menyimpan data *play store* dan tabel *appstoreapplication* untuk menyimpan data dari *app store*. *Data Dictionary* pada tabel *playapplication* dapat dilihat pada Tabel 4.1 berikut:

Tabel 4.1 Data dictionary pada tabel playapplications

Table	Column	Xtype	Data Type	Len	Null	Description
Playapplications	Id	Pk,fk	Int	11	N	Primary key untuk tabel applications
Playapplications	Title		Varchar	100	N	Nama dari aplikasi
Playapplications	Summary		Varchar	100	N	Outline/abstrak/tag dari aplikasi
Playapplications	Icon		Varchar	500	N	Gambar/logo/symbol dari aplikasi
Playapplications	Price		Tinyint	1	N	Harga jual dari aplikasi
Playapplications	Mininstalls		Int	11	N	Aplikasi tidak memiliki harga jual (true/false)
Playapplications	Maxinstalls		Int	11	N	Instalasi minimal dari aplikasi
Playapplications	Score		Decimal	10, 2	N	Instalasi maksimal dari aplikasi
Playapplications	Reviews		Int	11	N	Rata-rata skor rating dari aplikasi

Playapplications	Developer		Varchar	42	N	Jumlah review yang diberikan untuk aplikasi
Playapplications	Developeremail		Varchar	45	N	Nama dari pengembang aplikasi
Playapplications	Developerwebsite		Varchar	45	N	Alamat email milik pengembang aplikasi
Playapplications	Update		Varchar	45	N	Alamat situs milik pengembang aplikasi
Playapplications	Genre		Varchar	45	N	Tanggal terakhir aplikasi diperbarui
Playapplications	Genreid		Varchar	45	N	Kategori dari aplikasi
Playapplications	Description		Text		N	Id pada googleplay untuk kategori aplikasi
Playapplications	Descriptionhtml		Text		N	Deskripsi/gambaran menyeluruh dari aplikasi
Playapplications	Offersiap		Tinyint	1	N	Deskripsi/gambaran menyeluruh dari aplikasi versi html
Playapplications	Adsupported		Tinyint	1	N	Aplikasi menyediakan pembelian di dalam aplikasi (in app purchase)(true/false)
Playapplications	Androidversiontext		Varchar	45	N	Terdapat iklan dalam aplikasi

Playapplications	Androidversion		Varchar	10	N	Versi android yang dibutuhkan agar aplikasi dapat diinstall dalam bentuk text pada googleplay
Playapplications	Contentrating		Varchar	10	N	Versi android yang dibutuhkan agar aplikasi dapat diinstall
Playapplications	Video		Varchar	100	N	Rating usia untuk pengguna dari aplikasi
Playapplications	Recentchanges		Varchar	45	N	Link video dari aplikasi
Playapplications	Preregister		Tinyint	1	N	Gambaranperubahan yang terjadi pada versi aplikasi saat ini
Playapplications	Url		Varchar	100	N	Aplikasi dapat dicoba terlebih dahulu (true/false)
Playapplications	Appid		Varchar	100	N	Link aplikasi dapat diunduh pada googleplay

Data Dictionary pada tabel `googleplay_scrap_comments` dapat dilihat pada Tabel 4.2 berikut:

Tabel 4.2 Data dictionary pada tabel `googleplay_scrap_comments`

Table	Column	Xtype	Data Type	Len	Null	Description
<code>googleplay_scrap_comments</code>	Id	Pk,fk	Int	11	N	Primary key untuk tabel <code>googleplay_scrap_comments</code>
<code>googleplay_scrap_comments</code>	Comments		Text		N	Komentar aplikasi yang ada di play store

Data Dictionary pada tabel `googleplay_scrap_histograms` dapat dilihat pada Tabel 4.3 berikut :

Tabel 4.3 Data dictionary pada tabel `googleplay_scrap_histograms`

Table	Column	Xtype	Data Type	Len	Null	Description
<code>googleplay_scrap_histograms</code>	Id	Pk,fk	Int	11	N	Primary key untuk tabel <code>googleplay_scrap_histograms</code>
<code>googleplay_scrap_histograms</code>	Value		Varchar	254	N	Berisikan 5 skala penilaian aplikasi

Data Dictionary pada tabel `googleplay_scrap_permissions` dapat dilihat pada Tabel 4.4 berikut :

Tabel 4.4 Data dictionary pada tabel `googleplay_scrap_permissions`

Table	Column	Xtype	Data Type	Len	Null	Description
<code>googleplay_scrap_permissions</code>	Id	Pk,fk	Int	11	N	Primary key untuk tabel <code>googleplay_scrap_permissions</code>
<code>googleplay_scrap_permissions</code>	Permission		Varchar	45	N	Izin aplikasi dalam akses fitur di smartphone
<code>googleplay_scrap_permissions</code>	Description		Text		N	Keterangan izin aplikasi

Data Dictionary pada tabel `googleplay_scrap_reviews` dapat dilihat pada Tabel 4.5 berikut :

Tabel 4.5 Data dictionary pada tabel `googleplay_scrap_reviews`

Table	Column	Xtype	Data Type	Len	Null	Description
<code>googleplay_scrap_reviews</code>	Id	Pk,fk	Int	11	N	Primary key untuk tabel <code>googleplay_scrap_reviews</code> .
<code>googleplay_scrap_reviews</code>	Username		Varchar	45	N	Nama pengguna saat melakukan ulasan di aplikasi.
<code>googleplay_scrap_reviews</code>	Userimage		Varchar	100	N	Foto pengguna saat melakukan ulasan di aplikasi.

googleplay_scrap_reviews	Date		Varchar	50	N	Tanggal posting saat memposting ulasan di aplikasi yang bersangkutan.
googleplay_scrap_reviews	Url		Varchar	200	N	Tautan ulasan pengguna di halaman aplikasi.
googleplay_scrap_reviews	Score		Int	11	N	Penilaian aplikasi yang terhubung dengan ulasan.
googleplay_scrap_reviews	Title		Varchar	45	N	Judul ulasan saat memposting ulasan di aplikasi yang bersangkutan.
googleplay_scrap_reviews	Text		Text		N	Isi ulasan di aplikasi.

Data Dictionary pada tabel googleplay_scrap_screenshots dapat dilihat pada Tabel 4.6 berikut :

Tabel 4.6 Data dictionary pada tabel googleplay_scrap_screenshots

Table	Column	Xtype	Data Type	Len	Null	Description
googleplay_scrap_screenshots	Id	Pk,fk	Int	11	N	Primary key untuk tabel googleplay_scrap_permissions
googleplay_scrap_screenshots	Url		Text		N	Kumpulan <i>screenshot</i> di aplikasi

Data Dictionary pada tabel appstoreapplication dapat dilihat pada Tabel 4.7 berikut :

Tabel 4.7 Data dictionary pada tabel appstoreapplications

Table	Column	Xtype	Data Type	Len	Nu ll	Description
Appstoreapplications	Id_app (pk)	Pk,fk	Int	11	N	Primary key dan id untuk tabel applications
Appstoreapplications	Id_as		Varchar	15	N	Id aplikasi yang fungsinya mirip appid
Appstoreapplications	Appid		Varchar	100	N	Id aplikasi yang diberikan oleh googleplay untuk aplikasi
Appstoreapplications	Title		Varchar	100	N	Judul aplikasi
Appstoreapplications	Url		Varchar	100	N	Link aplikasi dapat diunduh pada appstore
Appstoreapplications	Descripti on		Text		N	Deskripsi/gambaran menyeluruh dari aplikasi

Appstoreapplications	Icon		Varchar	500	N	Logo dari aplikasi
Appstoreapplications	Genres		Varchar	254	N	Kategori yang terkait dengan aplikasi
Appstoreapplications	Genreids		Varchar	45	N	Kode kategori yang terkait dengan aplikasi
Appstoreapplications	Primarygenre		Varchar	45	N	Kategori utama yang terkait dengan aplikasi
Appstoreapplications	Primarygenreid		Varchar	11	N	Kode kategori utama yang terkait dengan aplikasi
Appstoreapplications	Contentrating		Varchar	10	N	Rating usia untuk pengguna dari aplikasi
Appstoreapplications	Languages		Varchar	254	N	Bahasa utama yang digunakan pada aplikasi
Appstoreapplications	Size		Varchar	50	N	Ukuran file aplikasi
Appstoreapplications	Requiredosversion		Varchar	10	N	System operasi minimal yang dibutuhkan untuk

						menginstall aplikasi
Appstoreapplications	Released		Varchar	25	N	Tanggal peluncuran aplikasi
Appstoreapplications	Updated		Varchar	25	N	Tanggal terakhir aplikasi diperbarui
Appstoreapplications	Releasenotes		Text		N	Keterangan untuk fitur yang ada pada saat perilsan aplikasi
Appstoreapplications	Version		Varchar	10	N	Versi aplikasi pada saat data diambil
Appstoreapplications	Price		Varchar	10	N	Nominal harga dari aplikasi
Appstoreapplications	Currency		Varchar	5	N	Mata uang utama yang ada selama transaksi di dalam aplikasi
Appstoreapplications	Free		Tinyint	1	N	1 atau true jika gratis dan 0 atau false untuk membayar

Appstoreapplications	Developerid		Varchar	15	N	Id dari masing-masing pengembang aplikasi
Appstoreapplications	Developer		Varchar	45	N	Nama dari pengembang aplikasi
Appstoreapplications	Developerurl		Varchar	254	N	Alamat situs milik pengembang aplikasi di situs apple
Appstoreapplications	Developerwebsite		Varchar	45	N	Alamat situs milik pengembang aplikasi
Appstoreapplications	Score		Decimal	10,2	N	Rating dari aplikasi
Appstoreapplications	Reviews		Int	11	N	Jumlah review yang ada di aplikasi
Appstoreapplications	Currentversion score		Decimal	10,2	N	Score yang didapatkan pada saat versi terakhir dirilis

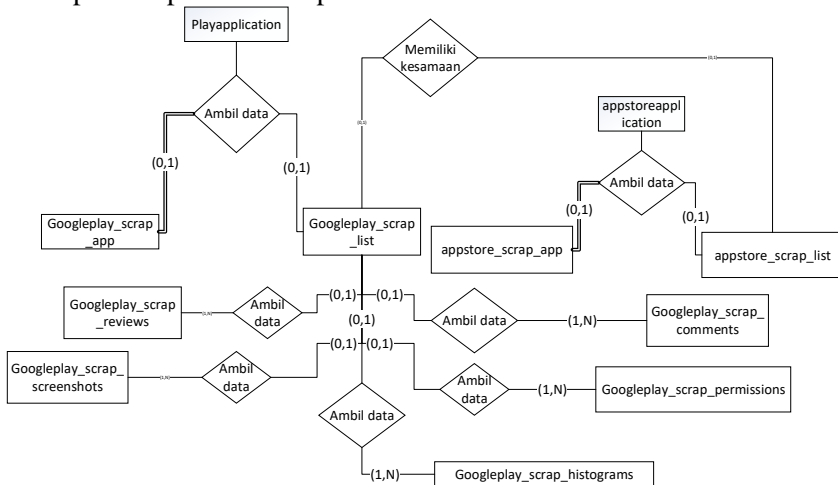
Appstoreapplications	Currentversion reviews		Varchar	10	N	Ulasan yang didapatkan pada saat versi terakhir dirilis
Appstoreapplications	Screenshots		Text		N	Contoh tampilan dari aplikasi yang ada di iphone
Appstoreapplications	Ipadcreenshots		Text		N	Contoh tampilan dari aplikasi yang ada di ipad
Appstoreapplications	Appletvscreenshots		Text		N	Contoh tampilan dari aplikasi yang ada di apple tv
Appstoreapplications	Supported devices		Text		N	Perangkat yang didukung oleh aplikasi

4.1.2.2 Perancangan Basis Data Logikal

Perancangan basis data logikal [12] adalah proses membangun suatu model informasi yang digunakan didalam perusahaan berdasarkan model data tersebut, tanpa memperhatikan pertimbangan penggunaan suatu DBMS dan perangkat keras lainnya. Tujuan dari perancangan ini adalah membangun model data logikal yang mengacu pada model data logikal dan juga memvalidasi model agar mendukung transaksi yang dibutuhkan oleh *user*.

4.1.2.2.1 Menentukan relasi untuk model data logikal

Pada tahapan ini bertujuan untuk membuat suatu relasi model data logika yang menggambarkan entitas, relasi, dan atribut yang telah diidentifikasi. Turunan dari tahapan ini terbagi menjadi 9 proses dalam menentukan relasi model data. Semula untuk hasil dari model data konseptual dapat dilihat seperti Gambar 4.22:

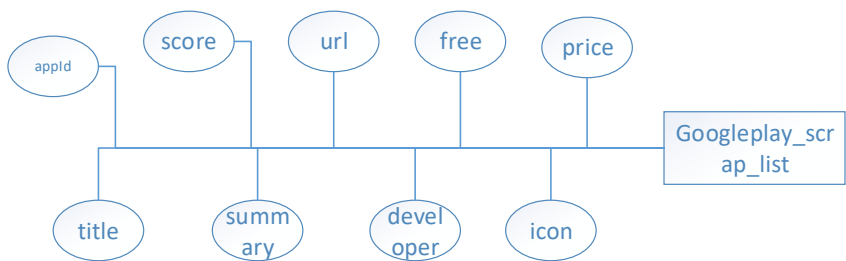


Gambar 4.22 model data konseptual

4.1.2.2.1.1 Mapping of Strong Entity Types

Dari proses mapping of strong entity types ini didapatkan hasil konversi diagram konseptual menjadi diagram logikal yang memiliki sifat entitas yang kuat, dalam hal ini memiliki primary key.

- Entitas Googleplay_scrap_list pada Gambar 4.23



Gambar 4.23 Entitas googleplay_scrap_list

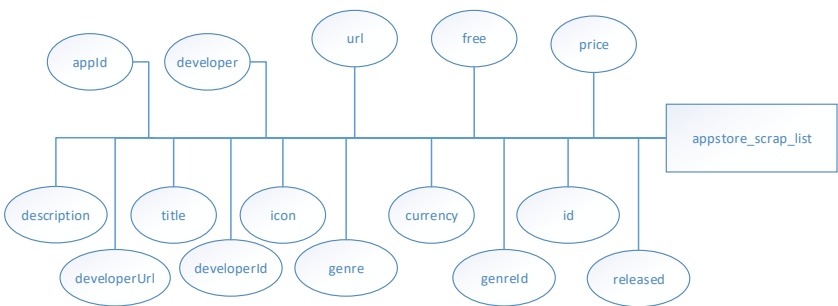
Entitas Googleplay_scrap_list dalam MRD ditunjukkan pada Gambar 4.24 berikut:

googleplay_scrap_list

appld	title	summary	icon	price	free	score	developer	url
-------	-------	---------	------	-------	------	-------	-----------	-----

Gambar 4.24 Mapping ERD googleplay_scrap_list

- Entitas appstore_scrap_list pada Gambar 4.25



Gambar 4.25 Entitas appstore_scrap_list

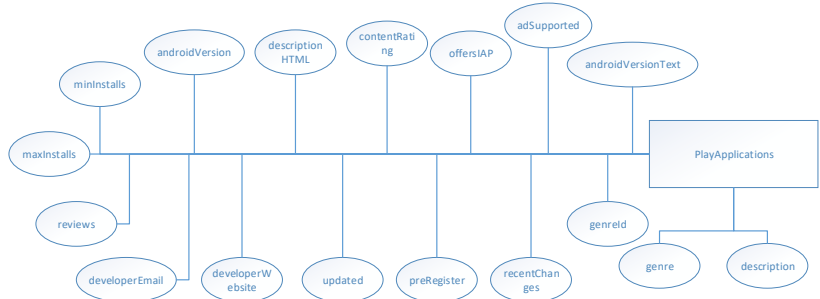
Entitas appstore_scrap_list dalam MRD ditunjukkan pada Gambar 4.26 berikut:

appstore_scrap_list

id	appld	title	description	icon	price	free	currency	developer	url	developerUrl	developerId	genre	genreId	released
----	-------	-------	-------------	------	-------	------	----------	-----------	-----	--------------	-------------	-------	---------	----------

Gambar 4.26 Mapping ERD appstore_scrap_list

- Entitas PlayApplications pada Gambar 4.27

**Gambar 4.27 Entitas PlayApplications**

Entitas playApplications dalam MRD ditunjukkan pada Gambar 4.28 berikut:

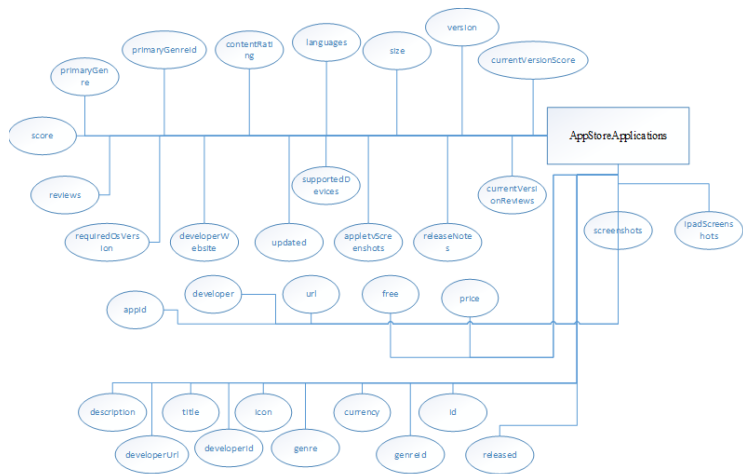
PlayApplications

description	androidVersion	minInstalls	maxInstalls	descriptionHTML	contentRating	offersIAP	adSupported	androidVersionText	reviews
developerEmail	genreId	developerWebsite	updated	preRegister	recentChanges	genre	appld		

Gambar 4.28 Mapping ERD PlayApplications

- Entitas AppStoreApplications pada Gambar 4.29

Entitas AppStoreApplications dalam MRD ditunjukkan pada Gambar 4.30 berikut:



Gambar 4.29 Entitas AppStoreApplications

AppStoreApplications

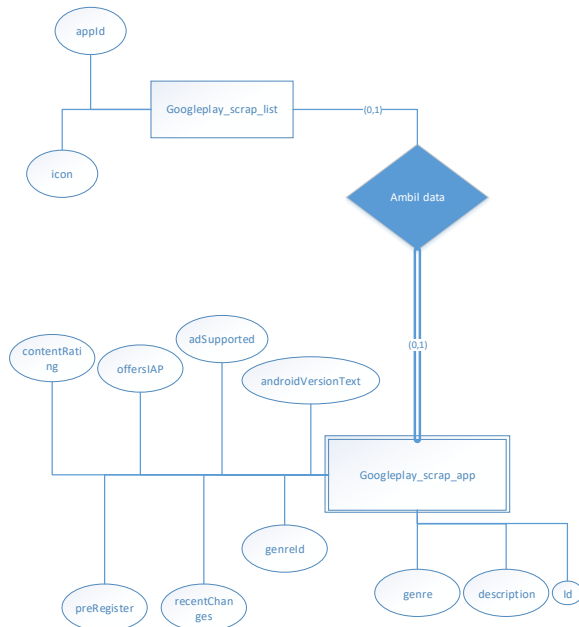
id	appId	title	url	description	icon	genres	genreIds	primaryGenre	primaryGenreId	contentRating	languages
size	requiredOsVersion	released	updated	releasedNotes	version	price	currency				
free	developerId	developerWebsite	score	reviews	screenshots						

Gambar 4.30 Mapping ERD AppStoreApplications

4.1.2.2.1.2 Mapping of Weak Entity Types

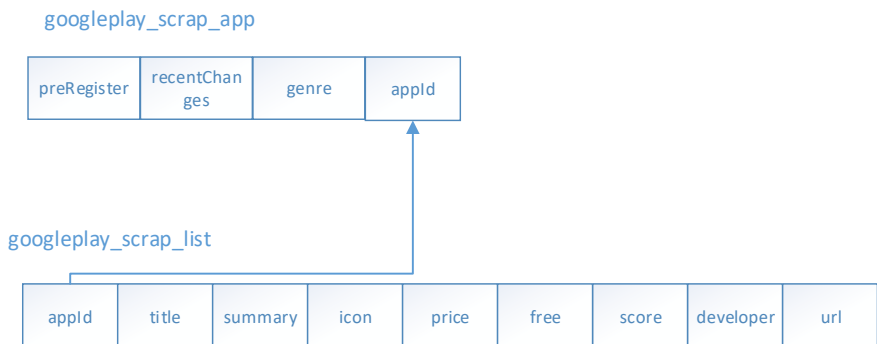
Dari proses *mapping of weak entity types* ini didapatkan hasil konversi diagram konseptual menjadi diagram logikal yang memiliki sifat entitas yang lemah, dalam hal ini perlu mengacu ke entitas yang kuat.

- Entitas `Googleplay_scrap_app` pada Gambar 4.31



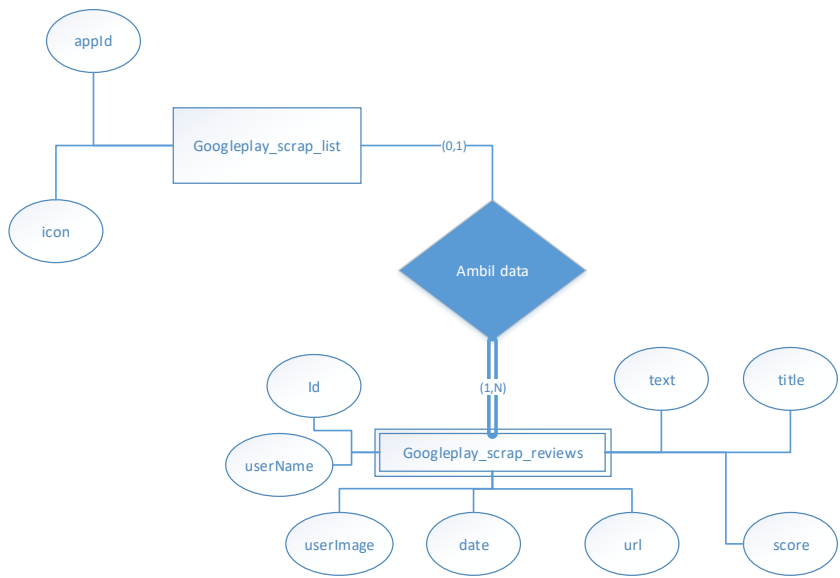
Gambar 4.31 Weak Entity pada googleplay_scrap_app

Weak Entity dari Googleplay_scrap_app yang berelasi dengan entitas Googleplay_scrap_list dalam MRD ditunjukkan pada Gambar 4.32 berikut:



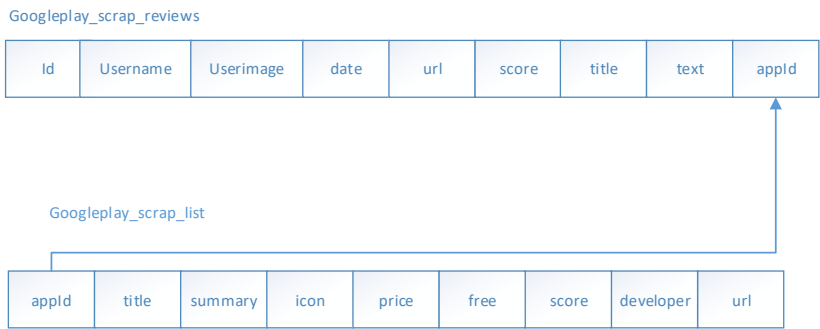
Gambar 4.32 Penambahan appId di googleplay_scrap_app

- Entitas googleplay_scrap_reviews pada Gambar 4.33



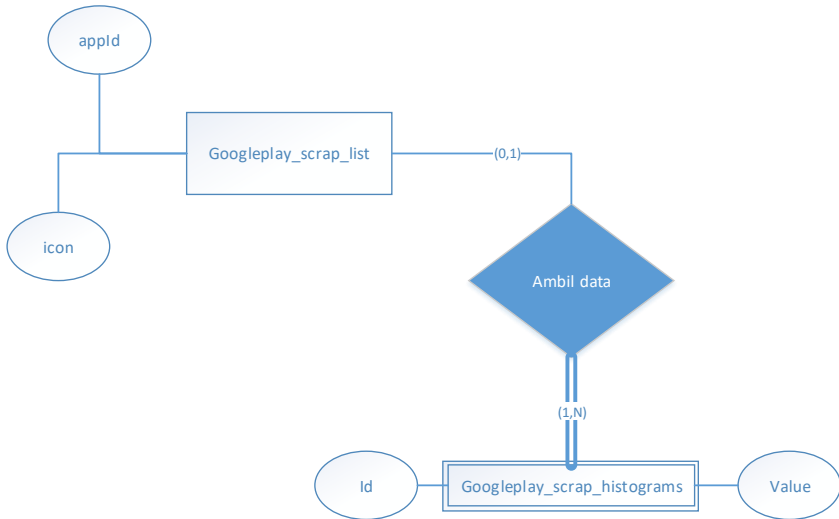
Gambar 4.33 Weak Entity pada googleplay_scrap_reviews

Weak Entity dari `Googleplay_scrap_reviews` yang berelasi dengan entitas `Googleplay_scrap_list` dalam MRD ditunjukkan pada Gambar 4.34 berikut:



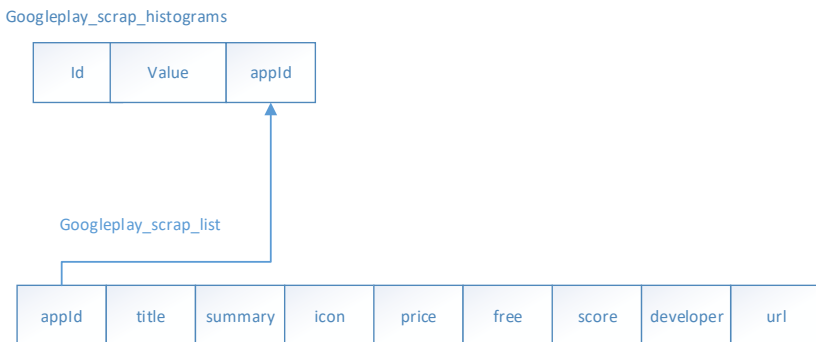
Gambar 4.34 Penambahan appId di googleplay_scrap_reviews

- Entitas `googleplay_scrap_histograms` pada Gambar 4.35



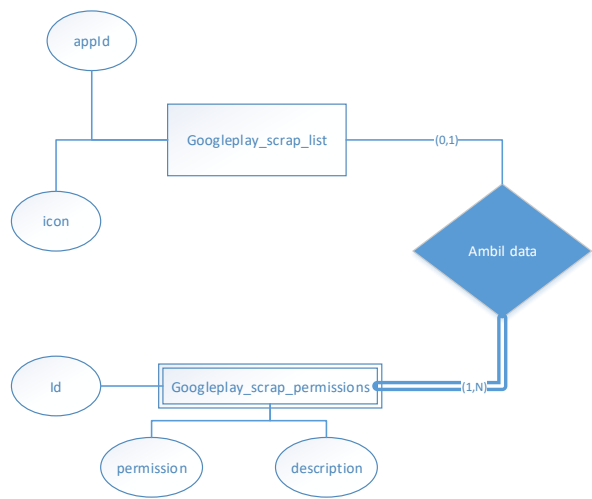
Gambar 4.35 Weak Entity pada googleplay_scrap_histograms

Weak Entity dari **Googleplay_scrap_histograms** yang berelasi dengan entitas **Googleplay_scrap_list** dalam MRD ditunjukkan pada Gambar 4.36 berikut:



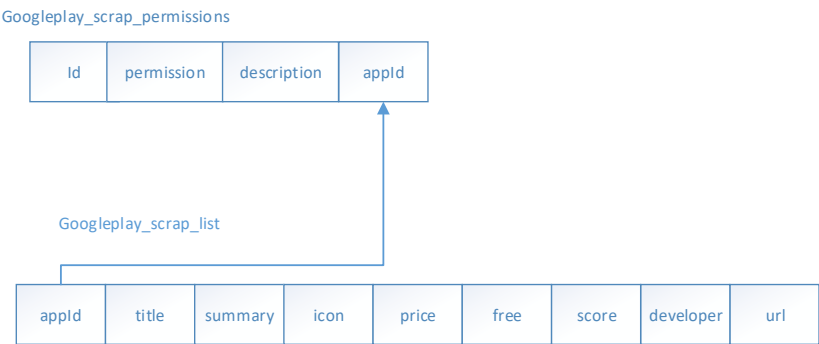
Gambar 4.36 Penambahan appId di googleplay_scrap_histograms

- Entitas **googleplay_scrap_permissions** pada Gambar 4.37



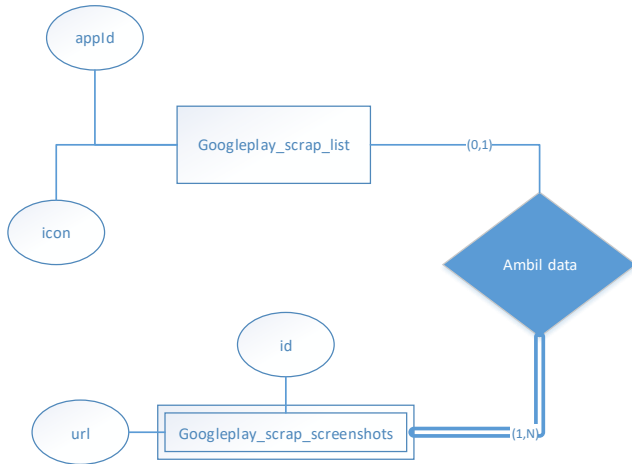
Gambar 4.37 Weak Entity pada googleplay_scrap_permissions

Weak Entity dari `Googleplay_scrap_permissions` yang berelasi dengan entitas `Googleplay_scrap_list` dalam MRD ditunjukkan pada Gambar 4.38 berikut:



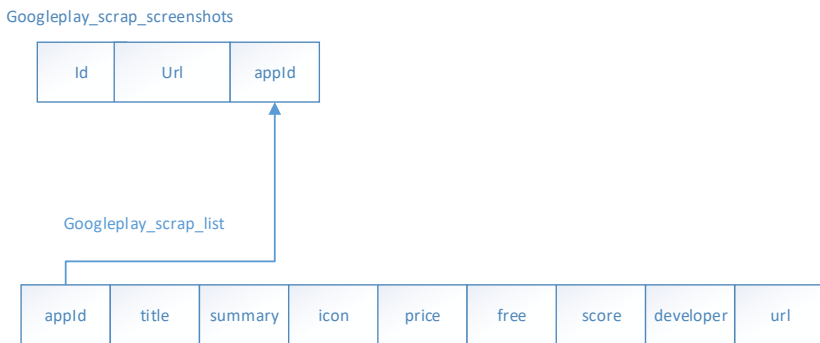
Gambar 4.38 Penambahan appId di googleplay_scrap_permissions

- Entitas `googleplay_scrap_screenshots` pada Gambar 4.39



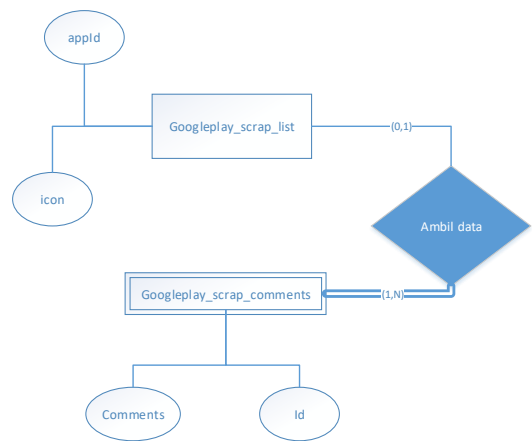
Gambar 4.39 Weak Entity pada googleplay_scrap_screenshots

Weak Entity dari Googleplay_scrap_screenshots yang berelasi dengan entitas Googleplay_scrap_list dalam MRD ditunjukkan pada Gambar 4.40 berikut:



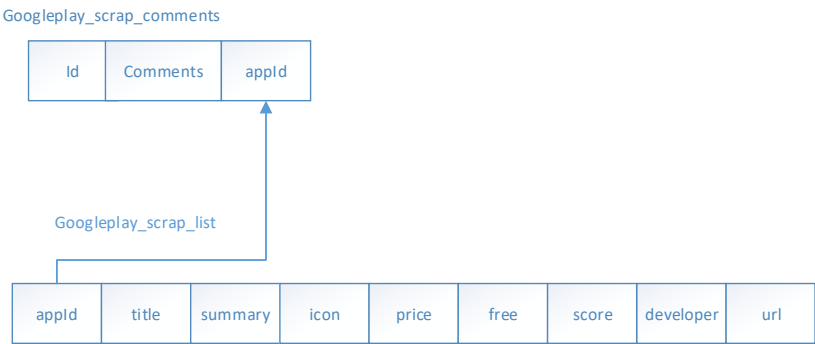
Gambar 4.40 Penambahan appId di googleplay_scrap_screenshots

- Entitas googleplay_scrap_comments pada Gambar 4.41



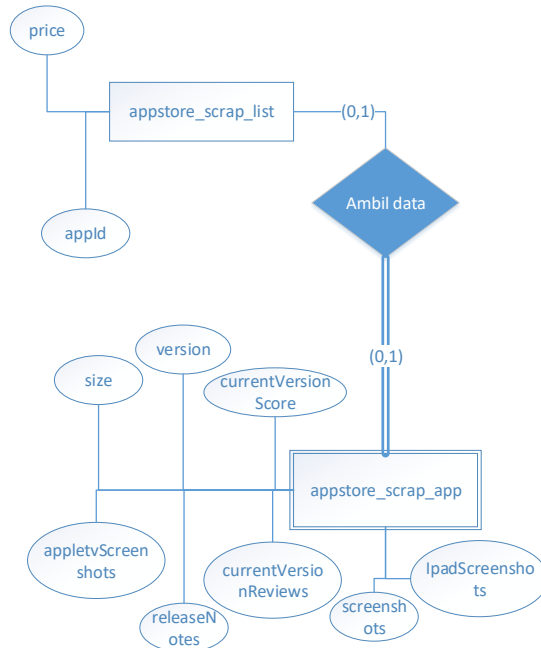
Gambar 4.41 Weak Entity pada googleplay_scrap_comments

Weak Entity dari Googleplay_scrap_comments yang berelasi dengan entitas Googleplay_scrap_list dalam MRD ditunjukkan pada Gambar 4.42 berikut:



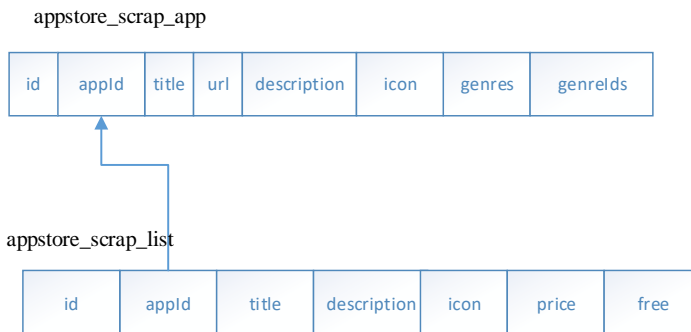
Gambar 4.42 Penambahan appId di googleplay_scrap_comments

- Entitas appstore_scrap_app pada Gambar 4.43



Gambar 4.43 Weak Entity pada appstore_scrap_app

Weak Entity dari appstore_scrap_app yang berelasi dengan entitas appstore_scrap_list dalam MRD ditunjukkan pada Gambar 4.44 berikut:



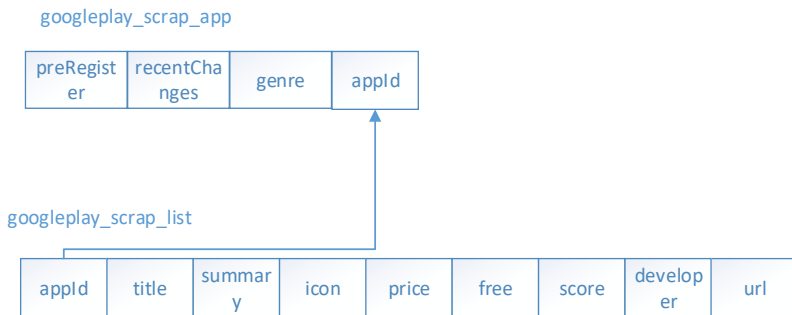
Gambar 4.44 Penambahan appId di appstore_scrap_app

4.1.2.2.1.3 *One-to-one (1:1) tipe relasi binary*

Proses *one-to-one* (1:1) tipe relasi *binary* ini hanya bisa berpasangan terhadap satu pasangan lainnya. Tujuan pada proses ini yaitu menentukan sifat relasi ini agar menghindari data yang redundan.

- Entitas `googleplay_scrap_list` dengan entitas `googleplay_scrap_app`

Entitas `googleplay_scrap_list` yang berelasi *one to one* dengan entitas `googleplay_scrap_app` dalam MRD ditunjukkan pada Gambar 4.45 berikut:



Gambar 4.45 relasi 1:1 `googleplay_scrap_app` dengan `googleplay_scrap_list`

- Entitas `appstore_scrap_list` dengan entitas `appstore_scrap_app`

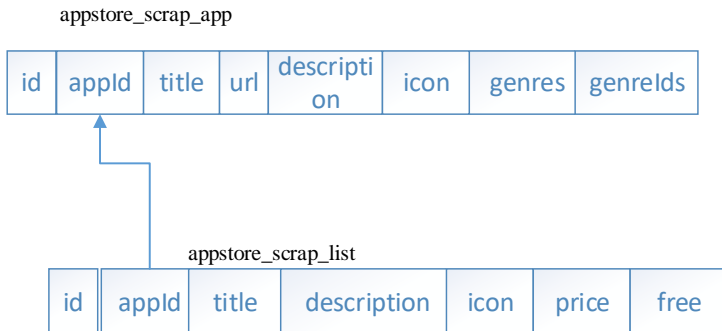
Entitas `appstore_scrap_list` yang berelasi *one to one* dengan entitas `appstore_scrap_app` dalam MRD ditunjukkan pada Gambar 4.46 berikut:

4.1.2.2.1.4 *One-to-many (1:*) tipe relasi binary*

Tidak dilakukan karena dalam ERD tidak ada atribut yang tipe relasi *binary* 1:N.

4.1.2.2.1.5 *One-to-one (1:1) tipe relasi rekursif*

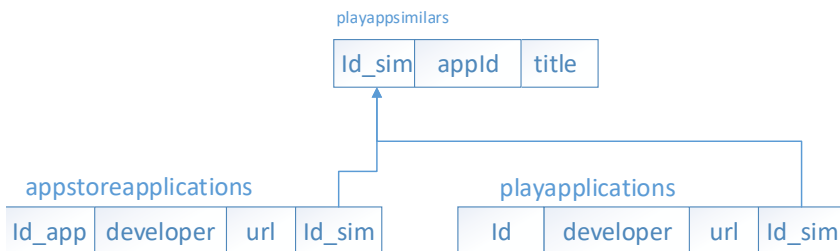
Tidak dilakukan karena dalam ERD tidak ada atribut yang tipe relasi rekursi 1:1.



Gambar 4.46 relasi 1:1 `appstore_scrap_app` dengan `appstore_scrap_list`

4.1.2.2.1.6 *Superclass* atau *subclass* tipe relasi

Relasi *superclass* pada tahapan ini terjadi antara entity `playapplications` dan `appstoreapplications`, relasi di antara dua entitas tersebut memiliki sifat *mandatory nondisjoint*. *Mandatory nondisjoint* adalah memiliki sifat dua entitas pembeda yang berada di bagian *subclass*. Super class disini berada di bagian relasi, dan hal itu melebur menjadi entitas baru dan menjadi jembatan antara dua *subclass*. Entitas baru ini bisa dilihat pada Gambar 4.47 berikut:



Gambar 4.47 penambahan entitas `playappsimilars`

Pada gambar, entitas baru ini diberi nama `playappsimilars` yang memiliki `id_sim` sebagai primary key super class, dan memiliki foreign key `id_sim` di subclassnya.

4.1.2.2.1.7 *Many-to-many* (*:*) tipe relasi *binary*

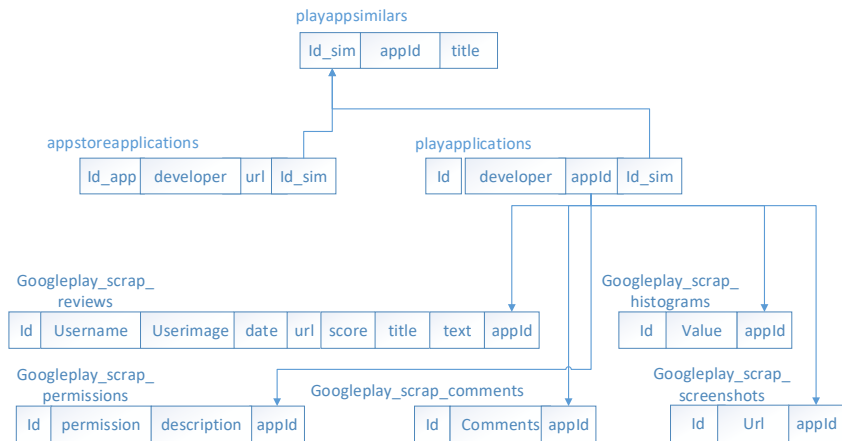
Tidak dilakukan karena dalam ERD tidak ada atribut yang tipe relasi *binary* M:N.

4.1.2.2.1.8 Tipe relasi kompleks

Tidak dilakukan karena dalam ERD tidak ada atribut yang memiliki relasi kompleks.

4.1.2.2.1.9 Atribut *multivalued*

Tidak dilakukan karena dalam ERD tidak ada atribut yang *multivalued*. Berdasarkan pemaparan diatas dapat disimpulkan bahwa hasil akhir untuk model data logikal dapat terlihat pada Gambar 4.48 dibawah ini:



Gambar 4.48 Hasil akhir model data logikal

4.1.2.3 Perancangan Basis Data Pisikal

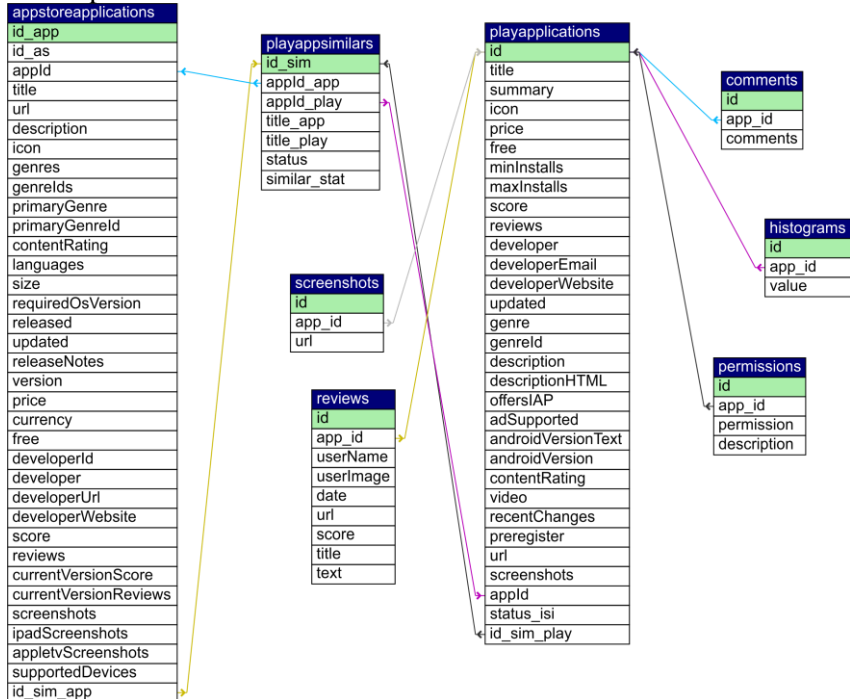
Perancangan basis data fisik [12] adalah proses pembuatan deskripsi implementasi dari basis data pada *media secondary storage*. Perancangan ini menggambarkan relasi dasar organisasi file pengindeksan yang digunakan untuk memperoleh akses yang efisien pada data, serta batasan integritas dan ukuran yang berhubungan. Langkah-langkah dalam metodologi perancangan basis data fisik adalah sebagai berikut:

4.1.2.3.1 Menterjemahkan model data logikal ke target DBMS

Pada tahapan ini bertujuan untuk menghasilkan skema basis data relasional dari model data logikal yang dapat diimplementasikan kedalam suatu DBMS. Aktivitas yang dilakukan meliputi merancang relasi dasar, merancang representasi dari data turunan, dan merancang *general constraints*.

4.1.2.3.1.1 Merancang relasi dasar

Pada tahapan ini, dalam mentranslasi model data logikal pada bagian relasi, perlu penyesuaian pada atribut dan relasinya. Dalam hal ini, atribut yang ada bisa dilihat di data dictionary, lalu untuk relasi bisa dilihat pada Gambar 4.49 berikut:



Gambar 4.49 Desain basis data di mysql

Relasi yang ada yaitu id_sim dimana primary key berada di playappsimilars, lalu relasi selanjutnya adalah app_id dimana primary key yaitu id yang berada di playapplications.

4.2 Rancang Bangun Prototipe

Rancang bangun prototipe bertujuan untuk membuat *draft* antar muka pengguna yang merepresentasikan berbagai fungsi yang telah direncanakan pada tahap selanjutnya. Luaran dari tahapan perancangan prototipe adalah *mockup* web aplikasi berbasis PHP/HTML yang mampu menyajikan software direktori yang terstruktur kepada pengguna perangkat lunak.

4.2.1 Requirement Gathering

Pencarian kebutuhan terhadap keberlangsungan pembuatan aplikasi sebagai tahap awal dalam pembuatan yang data hasil pencarian dapat membantu pembuatan aplikasi agar nantinya dapat disukai pengguna dan nyaman saat pemakaiannya. Pencarian kebutuhan dibagi menjadi dua yaitu ke *developer* dan pengguna dimana *developer* menggunakan metode wawancara yang mengarah ke kualitatif, lalu pengguna menggunakan metode kuisisioner yang mengarah ke kuantitatif.

Pencarian kebutuhan ke *developer* dilakukan minimal ke dua orang *developer* yang pertanyaan-nya mengarah ke publikasi *developer* selama ini. Pencarian kebutuhan ke pengguna dilakukan secara kuesioner dengan target minimal 30 pengguna yang pertanyaan-nya mengarah ke bagaimana data tersebut ditampilkan.

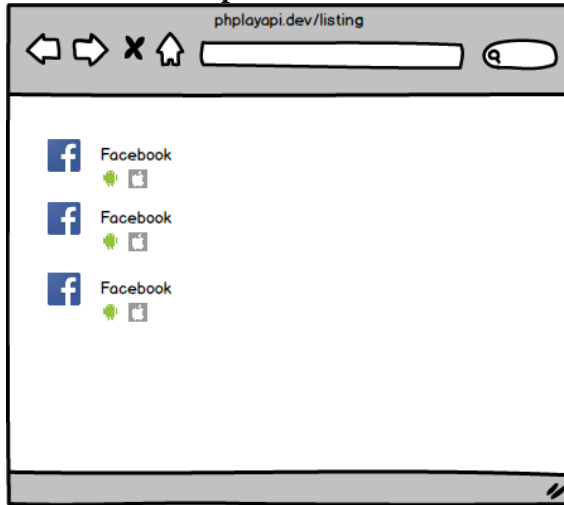
Hasil pencarian kebutuhan untuk *developer* didapati bahwa *developer* itu tidak menginginkan aplikasinya untuk di taruh di toko aplikasi pihak ketiga karena calon pengguna lebih banyak membuka *play store* dan pengguna lebih sering mengakses *play store*. Hal ini dapat disimpulkan toko aplikasi pihak ketiga membutuhkan metadata yang banyak dan melakukan *scraping* secara rutin.

Hasil pencarian kebutuhan untuk pengguna didapati bahwa terdapat 41 pengguna yang mengisi kuesioner dengan pengambilan hasil diatas 60% didapatkan hasil untuk penampilan halaman utama, yaitu judul, *icon*, dan rating. Lalu untuk penampilan halaman detail aplikasi, yaitu judul, deskripsi, rating, ukuran aplikasi, OS minimal yang diperlukan untuk melakukan instalasi, versi aplikasi, harga aplikasi, total *review* aplikasi, dan *screenshot* aplikasi.

4.2.2 Antar Muka Pengguna

Berikut ini adalah desain antar muka pengguna aplikasi. Antar muka yang dibuat terdiri dari dua antar muka, halaman pencarian dan halaman hasil pencarian. Untuk setiap antar muka pengguna yang dibuat, akan dijadikan acuan dalam pengembangan *use case*.

4.2.2.1 Halaman Daftar Aplikasi

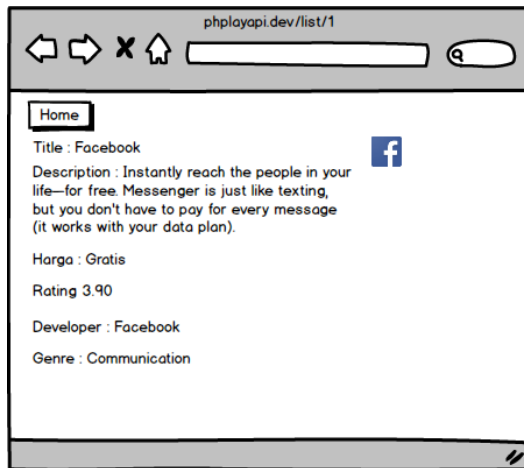


Gambar 4.50 Antar muka halaman utama aplikasi

Antar muka halaman utama aplikasi menampilkan daftar aplikasi yang berupa gabungan aplikasi di play store dan *app store* yang bisa dilihat pada Gambar 4.50. Pengguna bisa mengetahui ketersediaan aplikasi dengan melihat logo dibawah judul aplikasi. Apabila hanya tersedia logo android maka aplikasi hanya tersedia di *play store*, begitu juga sebaliknya.

4.2.2.2 Halaman Detail Aplikasi

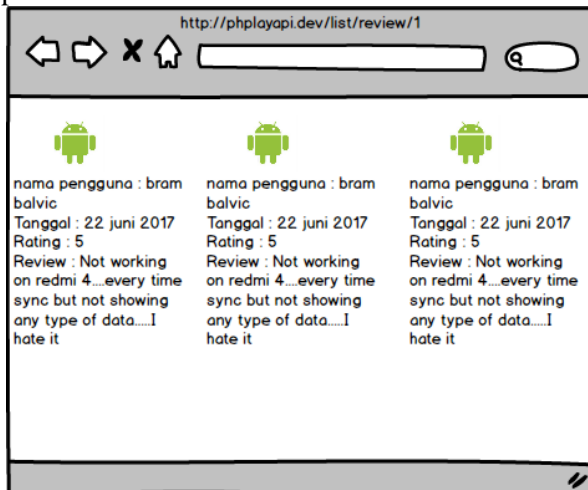
Antar muka halaman detail aplikasi menampilkan detail informasi aplikasi yang sebelumnya sudah dipilih di halaman utama aplikasi yang bisa dilihat pada Gambar 4.51. Pada halaman ini juga terdapat tangkapan layar dari aplikasi agar pengguna mendapatkan gambar dari aplikasi tersebut.



Gambar 4.51 Antar muka halaman detail aplikasi

4.2.2.3 Halaman Ulasan Aplikasi

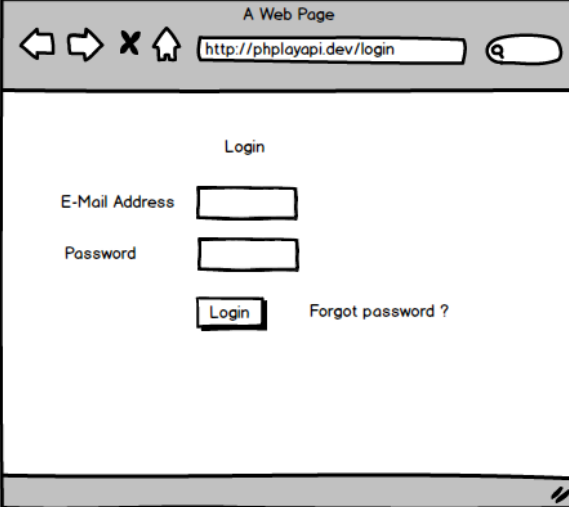
Antar muka halaman ulasan aplikasi yang bisa dilihat pada Gambar 4.52 menampilkan kumpulan ulasan dari sebuah aplikasi yang sebelumnya sudah dipilih dari halaman detail aplikasi. Pada halaman ini berisikan foto pengguna, nama pengguna, tanggal posting ulasan, penilaian terhadap aplikasi dan ulasan itu sendiri.



Gambar 4.52 Antar muka halaman ulasan aplikasi

4.2.2.4 Halaman *login* dan *register*

Antar muka halaman *login* dan *register* yang bisa dilihat pada Gambar 4.53 dan Gambar 4.54 menampilkan proses *login* dan registrasi pengguna yang ingin mendaftarkan aplikasinya.



A Web Page

http://phplayapi.dev/login

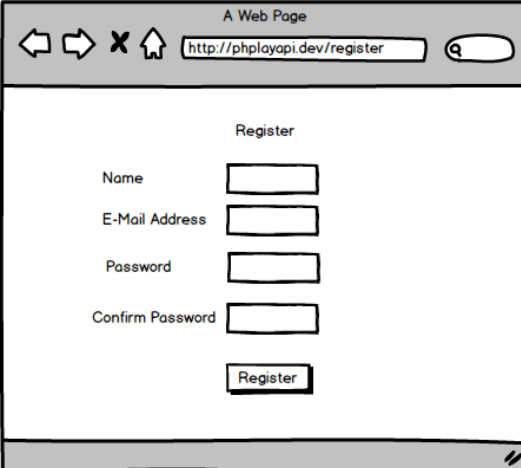
Login

E-Mail Address

Password

[Forgot password ?](#)

Gambar 4.53 Antar muka halaman login



A Web Page

http://phplayapi.dev/register

Register

Name

E-Mail Address

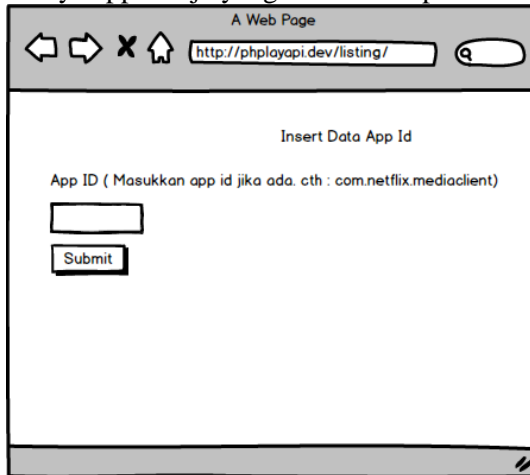
Password

Confirm Password

Gambar 4.54 Antar muka halaman pendaftaran akun

4.2.2.5 Halaman *Insert data*

Antar muka halaman untuk memasukkan data menampilkan proses yang terbagi dalam dua versi. Versi pertama yaitu memasukkan semua data aplikasi yang bisa dilihat pada Gambar 4.55, versi kedua yaitu memasukkan hanya appId saja yang bisa dilihat pada Gambar 4.56.



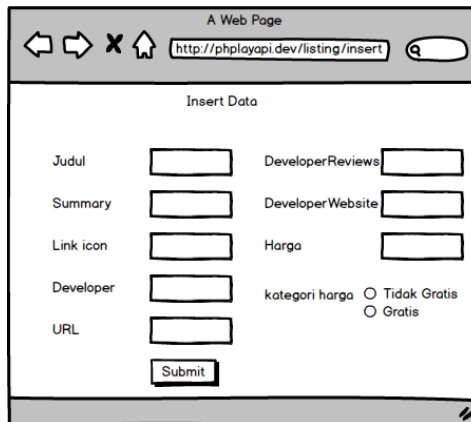
A Web Page

http://phplayapi.dev/listing/

Insert Data App Id

App ID (Masukkan app id jika ada. cth : com.netflix.mediaclient)

Gambar 4.55 Antar muka halaman insert data appId



A Web Page

http://phplayapi.dev/listing/insert

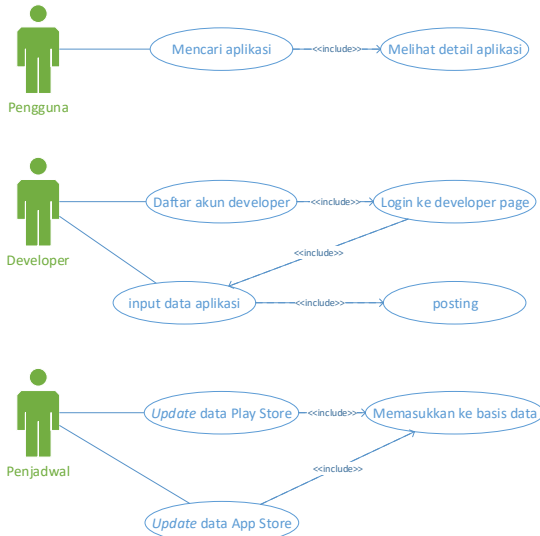
Insert Data

Judul	<input type="text"/>	DeveloperReviews	<input type="text"/>
Summary	<input type="text"/>	DeveloperWebsite	<input type="text"/>
Link icon	<input type="text"/>	Harga	<input type="text"/>
Developer	<input type="text"/>	kategori harga	<input type="radio"/> Tidak Gratis <input type="radio"/> Gratis
URL	<input type="text"/>		

Gambar 4.56 Antar muka halaman insert data lengkap

4.2.3 Use Case

Pada tahapan ini dibuat rancangan aplikasi melalui proses-proses di use case yang mengacu pada antar muka dan rancangan arsitektur sistem. Berikut ini adalah Gambar 4.57 yang menggambarkan use case secara keseluruhan.



Gambar 4.57 Use case

4.2.3.1 Daftar use case

Pada Tabel 4.8 menjelaskan tabel daftar *use case* berdasarkan aktor yang terkait pada aktivitas tertentu.

Tabel 4.8 Daftar Use Case

Kode	Aktor	Nama
UCP-1	Pengguna	Memilih aplikasi
UCP-2	Pengguna	Melihat detail aplikasi
UCD-1	Developer	Daftar akun developer
UCD-2	Developer	Login ke halaman developer
UCD-3	Developer	Input data aplikasi
UCD-4	Developer	Posting aplikasi
UCS-1	Penjadwal	Update data play store
UCS-2	Penjadwal	Update data app store
UCS-3	Penjadwal	Memasukkan data ke basis data

4.2.4 *Descriptive Case*

Descriptive case menjelaskan keterangan alur dari setiap *use case*. Setiap *use case* memiliki *normal flow* dan *alternate flow*.

Tabel 4.9 *descriptive case* memilih aplikasi

A. UCP-1 : Memilih Aplikasi
Normal Flow : User melakukan pemilihan produk dengan melihat daftar aplikasi yang ditampilkan di halaman utama.
Alternate Flow : Produk yang dicari kesulitan ditemui dikarenakan banyak aplikasi dan belum tentu ada di platform yang diinginkan.

Tabel 4.10 *descriptive case* melihat detail aplikasi

B. UCP-2 : Melihat detail aplikasi
Normal Flow : User mengklik aplikasi untuk melihat detail aplikasi pada halaman detail aplikasi. Selanjutnya, user akan diarahkan pada halaman detail aplikasi
Alternate Flow : -

Tabel 4. 11 *descriptive case* daftar akun developer

C. UCD-1 : Daftar akun developer
Normal Flow : Developer melakukan pendaftaran dengan mengisi form yang disediakan, setelah itu klik submit.
Alternate Flow : Email yang dimasukkan telah terdaftar, <i>text field</i> belum terisi semua sehingga gagal submit.

Tabel 4. 12 *descriptive case* login ke halaman developer

D. UCD-2 : Login ke halaman Developer
Normal Flow : Developer mengisi email dan password yang disediakan dalam <i>text field</i> lalu klik login
Alternate Flow : Gagal login dikarenakan password salah atau email tidak terdaftar

Tabel 4. 13 *descriptive case* input data aplikasi

E. UCD-3 : Input data aplikasi
Normal Flow : Developer mengisi data aplikasi melalui form yang disediakan lalu mengunggah tangkapan layar dan aplikasi. Setelah itu klik submit untuk memasukkan data yang sudah diisikan.
Alternate Flow : foto yang diunggah tak sesuai ketentuan, ekstensi foto yang diunggah bukan jenis gambar, foto yang diunggah melebihi ukuran yang sudah ditentukan sehingga mengakibatkan kegagalan, aplikasi yang diunggah tidak sesuai ketentuan dan ukuran yang ada, <i>text field</i> yang disediakan tidak diisikan semua sehingga menampilkan pesan error.

Tabel 4. 14 *descriptive case* posting aplikasi

F. UCD-4 : Posting aplikasi
Normal Flow : Aplikasi yang sudah diinputkan developer akan tampil di halaman utama aplikasi
Alternate Flow : Aplikasi tidak tampil dikarenakan data kurang lengkap atau masih dalam tahap penyaringan admin.

Tabel 4. 15 *descriptive case* update data play store

G. UCS-1 : Update data play store
Normal Flow : Program melakukan pengambilan data dari website API play store
Alternate Flow : Data tidak dapat diambil karena sudah dihapus pihak play store atau server di blokir play store sementara.

Tabel 4. 16 *descriptive case* update data app store

H. UCS-2 : Update data app store
Normal Flow : Program melakukan pengambilan data dari website API app store
Alternate Flow : Data tidak dapat diambil karena sudah dihapus pihak play store atau server di blokir play store sementara.

Tabel 4. 17 *descriptive case* memasukkan data ke basis data

A. UCS-3 : Memasukkan data ke basis data
Normal Flow : Data masuk ke tabel yang sudah ditentukan melalui proses <i>insert</i> apabila data tidak ada setelah di cek dan melakukan proses <i>update</i> apabila data yang sudah ada mengalami perubahan konten di dalamnya.
Alternate Flow : Data gagal masuk dikarenakan koneksi tidak stabil yang dalam hal ini variabel yang menampung data sementara kosong sehingga menyebabkan kegagalan.

BAB V

IMPLEMENTASI

Pada bab implementasi ini dijelaskan hasil implementasi yang telah dilakukan dalam tugas akhir terkait desain prototipe aplikasi, pengolahan data, dan implementasi aplikasi itu sendiri. Implementasi dilakukan dengan menggunakan Bahasa pemrograman PHP, dan penerapan validasi prototipe menggunakan metode *usability testing*.

5.1 Perangkat Penelitian

Pada Pengembangan aplikasi berbasis website ini dibutuhkan perangkat-perangkat yang dapat mengakomodir mulai dari pengambilan data, pengolahan data dalam mencari similarity aplikasi, hingga membuat prototipe aplikasi. Adapun perangkat yang dibutuhkan meliputi perangkat keras dan perangkat lunak dengan spesifikasi yang bisa dilihat pada Tabel 5.1 dan Tabel 5.2:

Tabel 5.1 Spesifikasi Perangkat Keras

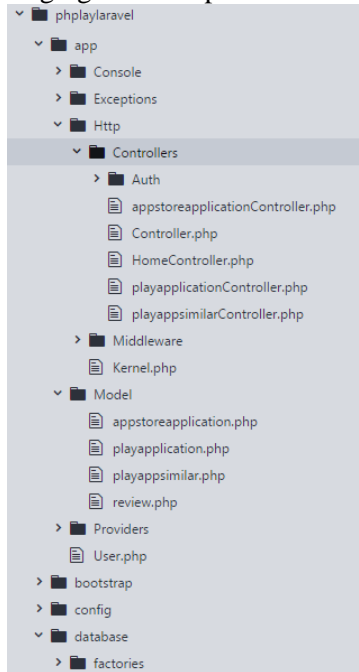
Perangkat Keras	Spesifikasi
Laptop	Processor : Intel® Core™ i7-6700HQ Processor 6M Cache, up to 3.50 GHz
	RAM 8.00 GB
	SSD 480 GB
	Windows 10 Pro

Tabel 5.2 Spesifikasi Perangkat Lunak

Perangkat Lunak	Spesifikasi
Database Server	MySQL
Bahasa	PHP 5.6.28
Framework	Laravel Framework 5.4.23
Editor	Brackets Release 1.9 build 1.9.0-17312
Web Server	Apache/2.4.23

5.2 Struktur Direktori

Pada Gambar 5.1 adalah struktur direktori untuk pengembangan prototipe aplikasi agregasi toko aplikasi.



Gambar 5.1 Struktur Direktori Aplikasi

5.3 Deployment Website API

Dalam melakukan pengambilan data, dilakukan pemasangan Website API ke virtual private server (VPS) melalui heroku.com, tujuan pemasangan demi memudahkan pengambilan data, menghemat bandwidth, dan mempersingkat waktu pengambilan. Proses pemasangan dengan melakukan instalasi Heroku CLI. Hal yang pertama dengan melakukan login ke akun heroku dengan command “heroku login”. Setelah melakukan login pindahkan direktori ke folder yang ingin dilakukan sinkronisasi ke heroku dengan command “cd”. Lalu

dilakukan clone yang berguna untuk mengcopy folder ke direktori heroku dengan nama project dengan command “heroku git:clone -a pstoreapi”. Setelah itu dilakukan pemasangan dengan command “git add .”. lalu dilakukan kesepakatan dengan command commit yaitu “git commit -am "make it better". Tahap terakhir yaitu melakukan pengiriman ke heroku dengan command “git push heroku master”. Hasil akhirnya akan seperti Gambar 5.2 berikut :

```
remote:       async@0.2.10 /tmp/build_fb5f5c9e49186f316e8451242940b1cf/no
remote:       source-map@0.5.6 /tmp/build_fb5f5c9e49186f316e8451242940b1c
ap
remote:       Installing any new modules (package.json)
remote:
remote:       -----> Caching build
remote:       Clearing previous node cache
remote:       Saving 2 cacheDirectories (default):
remote:       - node_modules
remote:       - bower_components (nothing to cache)
remote:
remote:       -----> Build succeeded!
remote:       -----> Discovering process types
remote:       Procfile declares types -> web
remote:
remote:       -----> Compressing...
remote:       Done: 17.8M
remote:       -----> Launching...
remote:       Released v6
remote:       https://pstoreapi.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/pstoreapi.git
   798dc30..9ec0848  master -> master
```

Gambar 5.2 Hasil akhir proses pemasangan website API

5.4 Pengambilan Data

Sebelum menampilkan data aplikasi, dilakukan pengambilan data dengan melakukan *web scraping*. *Web scraping* dilakukan dengan menggunakan *client* URL (cURL) yang mana data yang akan diambil berupa json. Json yang berhasil diambil lalu di *decode* dan untuk mendapatkan nilai yang ada di dalamnya, diperlukan wadah untuk meletakkan nilai dari json yang dalam hal ini yaitu variabel. Sebelum diletakkan di variabel, perlu diketahui nilai yang dibawa json. Json membawa 31 nilai dalam satu *array* yang diambil dari *play store* dan 33 nilai dalam satu *array* yang diambil dari *app store*. Variabel yang sudah

membawa nilai lalu diletakkan didalam *query* yang berfungsi memasukkan data ke dalam basis data.

Dijelaskan sebelumnya bahwa pengambilan data memiliki 2 tahap yang bersifat umum dan spesifik. Dua tahap itu membutuhkan koneksi yang menjadi jalur penghubung basis data dan variabel nilai yang menampung json.

Berikut adalah potongan kode koneksi ke basis data yang aksesnya menggunakan PDO.

```

1. <?php
2. $db_user = 'root';
3. $db_pass = '';
4. $db_name = 'googleplay_api';
5.
6. // connect to database
7. $dsn = 'mysql:dbname=' . $db_name . ';charset=utf8;host=localhost';
8. try {
9.     $dbh = new PDO( $dsn, $db_user, $db_pass
10. );
11. } catch ( PDOException $e ) {
12.     echo 'Koneksi Gagal : ' . $e-
13. >getMessage();
14. }
15. ?>

```

Gambar 5.3 Konfigurasi MySQL PDO

Kode pada Gambar 5.3 adalah konfigurasi untuk koneksi basis data yang menghubungkan ke basis data googleplay_api.

5.4.1 Pembuatan Scraper Play Store

Pada bagian ini akan menjelaskan proses *scraper* yang pada prosesnya akan mengambil data *list*, data *app*, data *review*, data *screenshot*, data *permission*, data *comment*, dan data *histogram*. *List* adalah proses mengambil daftar aplikasi dari suatu koleksi. *App* adalah proses mengambil data aplikasi secara utuh dari aplikasi. *List* dan *app* akan mengambil data di *play store* dan *app store*.

```

1. include('koneksi.php');
2. $query = "select category from category";
3. $result1 = $dbh->query($query)->fetchAll();
4. foreach($result1 as $r1){
5.     $category = $r1['category'];

```

Gambar 5.4 Kode untuk mengambil daftar kategori

Kode pada Gambar 5.4 adalah proses pengambilan daftar aplikasi yang dipilah-pilah menjadi berbagai kategori yang diambil di tabel *category*. Lalu setelah dari tabel *category* itu diambil kolom kategori yang berisikan 58 kategori.

```

1. $service_url="https://pstoreapi.herokuapp.com/list/new_paid/$category/110/en/en/100/100";
2. $curl = curl_init($service_url);
3. curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
4. curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 0);
5. curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);
6. //execute the session
7. $curl_response = curl_exec($curl);
8. //finish off the session
9. curl_close($curl);
10. $curl_jason = json_decode($curl_response, true);

```

Gambar 5.5 Kode untuk mendapatkan daftar aplikasi

Kode pada Gambar 5.5 merupakan lanjutan dari proses pengambilan data kategori dimana kategori yang ada dimasukkan ke proses *curl* untuk mendapatkan json-nya. Lalu json itu dilakukan perulangan yang hasilnya dapat ditaruh di dalam variabel untuk kemudian dimasukkan ke dalam basis data.

```

1. foreach ($curl_jason as $val) {
2.     if(isset($val['url'])){
3.         $url = htmlspecialchars($val['url'], ENT_QUOTES);
4.     }
5.     else{

```

```

6. $url = "";
7. }
8. if(isset($val['title'])){
9. $title = htmlspecialchars($val['title'], ENT_QUOTES);
10. }
11. else{
12. $title = "";
13. }
14. }

```

Gambar 5.6 Kode untuk mengkonversi karakter

Kode pada Gambar 5.6 adalah proses pengolahan data yang diambil dari curl dimana luaran nilai json tersebut tidak serta merta bersih datanya, sehingga harus dilakukan konversi karakter yang belum terdefinisi menjadi entitas HTML dengan *htmlspecialchars*.

```

1. $query = "select appId from playapplications where appId='".$appid."'";
2. $result1 = $dbh->query($query)->fetch();
3. if($result1==false){
4. $table = 'applications'; // ubah ke nama tabel
5. $field = `url`,`title`,`appId`,`summary`,`developer`,`icon`,`score`,`price`,`free`; // kolomnya, kalo > 1 pisahkan pakai koma
6. $val = '?,?,?,?,?,?,?,?,?'; // ini sesuai jumlah kolomnya, pakai koma
7. $array = array( $url, $title, $appid, $summary, $developer, $icon, $score, $price, $free ); // sesuai jumlah kolom juga
8. $sth = $dbh->prepare( "INSERT INTO $table ($field) VALUES ($val)" );
9. $input = $sth->execute( $array );
10. }
11. }

```

Gambar 5.7 Kode untuk memasukkan daftar aplikasi ke database

Kode pada Gambar 5.7 adalah proses pemasukan data dengan pengecekan kondisi dimana aplikasi yang sudah ada, tidak akan masuk lagi ke dalam basis data dengan mengecek appId yang nantinya akan dijadikan *unique key*.

Lalu proses selanjutnya adalah mengambil data spesifik aplikasi secara utuh. Proses yang membedakan antara *list* dan *app* adalah pemanggil untuk perulangannya tidak melalui kategori seperti halnya pada *list*, melainkan pada appId yang berfungsi mengambil data sisa yang tidak ada pada proses *list*. Lalu pembeda lainnya berada di data yang bersifat *array*, sehingga diperlukan pemilahan data.

```
1. $query = "select appid from playapplications";
2. $result1 = $dbh->query($query)->fetchAll();
3. foreach($result1 as $r1){
4.     $appid_play = $r1['appid'];
```

Gambar 5.8 Kode untuk mendapatkan appId

Kode pada Gambar 5.8 adalah proses pengambilan appId yang menjadi acuan dalam pengambilan data spesifik. appId diambil dari tabel playapplications.

```
1. if(isset($vali['recentChanges'])){
2.     $recentChanges = htmlspecialchars(implode
3.     (" ", $vali['recentChanges']), ENT_QUOTES);
4. }
5. else if(strlen($vali['recentChanges'])==0){
6.     $recentChanges = "";
7. } else {
8.     $video = htmlspecialchars($vali['recentCh
9.     anges'], ENT_QUOTES);
10. }
```

Gambar 5.9 Kode untuk memecah array dan mengkonversi karakter

Kode pada Gambar 5.9 adalah proses penyaringan data dengan melakukan konversi ke html *entities* dan memilah *array* tersebut dengan *implode*.

```

1. $query2 = "UPDATE playapplications SET minInstal
ls = '$minInstalls', maxInstalls = '$maxInstalls
', reviews = '$reviews', developerEmail = '$devel
operEmail', developerWebsite = '$developerWebsite
', updated = '$updated', genre = '$genre', genreI
d = '$genreId', description = '$description', des
criptionHTML = '$descriptionHTML', offersIAP = '$
offersIAP', video = '$video', adSupported = '$adS
upported', androidVersion = '$androidVersion', co
ntentRating = '$contentRating', androidVersionTex
t = '$androidVersionText', recentChanges = '$rece
ntChanges', preregister = '$preregister' WHERE ap
pId = '$appid_play '";
2.
3. if($dbh->exec($query2))
4. {
5.     echo 'sukses';
6. }
7. else {
8.     echo 'gagal';
9. }

```

Gambar 5.10 Kode untuk memasukkan data ke basis data dengan update

Kode pada Gambar 5.10 adalah memasukkan data aplikasi spesifik ke dalam tabel playapplications berdasarkan appId, menggunakan *update* sebagai cara memasukannya karena baris data sudah dibentuk di proses *list*.

5.4.2 Pembuatan *Scraper App Store*

Pembuatan *scraper app store* pada prosesnya sama seperti di sub bab 5.3.1 mengenai pembuatan *scraper* play store, namun yang membedakan dari segi pengambilan data acuan spesifik dimana terdapat dua pemanggilan yaitu id dan appId seperti dijelaskan pada potongan kode di Gambar 5.11

```

1. $query = "select id_as,appId from appstoreapplica
   tions where id_app >= 1";
2.     $result1 = $dbh->query($query)->fetchAll();
3.     foreach($result1 as $r1){
4.         $id_as = $r1['id_as'];
5.         $appId = $r1['appId'];

```

Gambar 5.11 Kode untuk mengambil id_as dan appId di appstore

Perbedaan lainnya yaitu dari segi penampung variabelnya yang disiapkan untuk masuk ke database. Hal ini dikarenakan kolom-kolom pada database isinya berbeda, namun pada pengolahannya tetap sama.

5.5 Perhitungan Similarity

Pada pengambilan data *play store* dan *app store* terdapat data yang sama antar tabel. Hal itu membutuhkan jembatan agar data yang ada pada 2 tabel tersebut mempunyai setidaknya beberapa data yang sama.

Untuk mencari kesamaan digunakan fungsi levenshtein pada mysql dengan mengambil appId, title, dan developer. Levenshtein membutuhkan ketiga atribut tersebut, salah satu diantaranya yaitu appId berupa *unique key*, sedangkan dua atribut lainnya sebagai pendukung dalam memperkuat pengolahan *similarity*.

Dikarenakan fungsi levenshtein sudah tertanam di mysql, maka pada kode di php hanya tinggal memanggil fungsinya saja.

```

1. ini_set('memory_limit', '2048M');
2. include('koneksi.php');
3. $query="SELECT * FROM playapplications where stat
   us_isi = '0' ORDER BY maxInstalls desc";
4. $result = $dbh->query($query)->fetchAll();

```

Gambar 5.12 Kode untuk mengambil data di playstore dengan kondisi

Kode pada Gambar 5.12 adalah untuk memanggil data dari tabel `playapplications` dimana jika belum dicari *similarity* akan memiliki status 0 dan berdasar `maxInstalls` yang berarti aplikasi dengan *download* terbanyak akan dicari kesamaannya terlebih dahulu. Hal ini bertujuan untuk mengefektifkan data yang ingin ditampilkan. Data-data yang ditampilkan dengan jumlah unduh lebih besar akan ditampilkan terlebih dahulu. Lalu pengaturan *memory limit* dengan ukuran dua *gigabyte* dikarenakan fungsi `levenshtein` memakan memori yang cukup besar ketika mencari kesamaan di `appstoreapplications` yang berjumlah 18336 data.

```

1. foreach($result as $r){
2.     $appid_play = "";
3.     $appId_app = "";
4.     $title_play = "";
5.     $appid_play = strtolower($r['appId']);
6.     $title_play = strtolower($r['title']);
7.     $developer_play = strtolower($r['developer'])
8.     ;
9.     $query2 = "select (levenshtein(?, appId)+levensht
10.     ein(?, title)+levenshtein(?, developer)) as jumla
11.     h ,appId, title, developer from `appstoreapplicat
12.     ions` ORDER BY `jumlah` ASC limit 1";
13.     $sth = $dbh->prepare($query2);
14.     $sth->execute([$appid_play,$title_play, $developer_pla
15.     y] );
16.     $sth->execute();
17.     $result2 = $sth->fetch(PDO::FETCH_ASSOC);
18.     $hasil = $result2['jumlah'];
19.     $appId_app = strtolower($result2['appId']);
20.     $title_app = strtolower($result2['title']);
21.     $developer_app = strtolower($result2['develop
22.     er']);

```

Gambar 5.13 Kode untuk menjalankan fungsi `levenshtein`

Kode pada Gambar 5.13 adalah proses mengubah semua nilai yang didapat pada Gambar 5.12 menjadi *lowercase* terlebih dahulu agar dapat diproses di `levenshtein`. Hasil `levenshtein`

selanjutnya dihitung dan diambil satu terbaik lalu diletakkan/disimpan di variabel jumlah.

```

1. if(($developer_app == $developer_play || $title_p
   lay == $title_app || $appid_play == $appId_app) &
   & $hasil < 30 ){
2.     echo '1';
3.     $status = '1';
4. }
5. } else if($hasil < 60){
6.     $status = '2';
7. }
8. } else {
9.     $status = '0';
10. }

```

Gambar 5.14 Kode untuk memasukkan data ke basis data dengan kondisi

Kode pada Gambar 5.14 menjelaskan proses hasil kalkulasi levenshtein dimana jika jumlah tiga perhitungan levenshtein tidak melebihi dari 30, maka akan dianggap aplikasi tersebut sama baik di *app store* maupun di *play store*. Semakin kecil nilai perhitungan levenshtein, maka tingkat kesamaan akan semakin tinggi. Perhitungan dengan hasil tidak boleh melebihi 30 berdasarkan hasil percobaan yang jika dinaikkan nilainya akan mengurangi tingkat kemungkinan bertemunya aplikasi yang sama. Jika nilai kurang dari 30 maka akan dinilai cocok dan memiliki status satu. jika kurang dari 60 maka tingkat kecocokan akan berkurang dan aplikasi yang didapatkan tidak akan sama dan kondisi ini akan diberi status dua. Jika tetap tidak menemui yang sama hingga melebihi 60 akan memiliki status nol.

5.6 Rancang Bangun Prototipe

Tahapan ini merupakan implementasi dari tahapan analisis dan perancangan ke dalam bentuk aplikasi sesuai dengan penjelasan lingkungan implementasi.

5.6.1 Implementasi Halaman Daftar Aplikasi

Halaman daftar aplikasi pada aplikasi ini berperan dalam menampilkan aplikasi-aplikasi yang tersedia di *play store* maupun *app store*. Konten yang disuguhkan dalam menampilkan daftar aplikasi berupa ikon, judul dan indikator aplikasi. Untuk melihat detail aplikasi, pengguna cukup klik judul aplikasi yang tersedia.

```

1. @foreach ($playapp as $r1)
2.     <div class="col-md-
4 agileinfo_new_products_grid agileinfo_new_produ
cts_grid_mobiles">
3.         <div class="agile_ecommerce_tab_left
mobiles_grid">
4.             <div>
5.                 
6.             </div>
7.             <h5><a href="list/{{ $r1-
>id_sim }}">{{ $r1->title_play }}</a></h5>
8.             <div class="simpleCart_shelfItem">
9.                 <?php $a = $r1->status; ?>
10.                 @if ($a=='1')
11.                     <p>Tersedia di <i>Android</i> d
an <i class="item_price">IOS</i></p>
12.                 @else
13.                     <p> <i class="item_price">And
roid</i></p>
14.                 @endif
15.             </div>
16.         </div>
17.     </div>
18. @endforeach

```

Gambar 5.15 Kode untuk menampilkan daftar aplikasi

Kode pada Gambar 5.15 digunakan untuk menampilkan daftar aplikasi serta menampilkan ketersediaan aplikasi di *play store* ataupun di *app store*.

```

1. use PlayAppSimilar\Model\playappsimilar;
2. public function index(){

```

```

3.     $playapp = playappsimilar::join('playapplicat
      ions', 'playappsimilar.appId_play', '=', 'playap
      plications.appId')-
      >join('appstoreapplications', 'playappsimilar.ap
      pId_app', '=', 'appstoreapplications.appId')-
      >select('playappsimilar.*', 'playapplications.*'
      , 'appstoreapplications.*')-
      >orderBy('playapplications.maxInstalls', 'desc')-
      >paginate(12);
4. return view('core/home', compact('playapp'));
5.   }

```

Gambar 5.16 kueri untuk menampilkan daftar aplikasi

```

6. Kode pada
   use PlayAppSimilar\Model\playappsimilar;
7. public function index(){
8.     $playapp = playappsimilar::join('playapplicat
      ions', 'playappsimilar.appId_play', '=', 'playap
      plications.appId')-
      >join('appstoreapplications', 'playappsimilar.ap
      pId_app', '=', 'appstoreapplications.appId')-
      >select('playappsimilar.*', 'playapplications.*'
      , 'appstoreapplications.*')-
      >orderBy('playapplications.maxInstalls', 'desc')-
      >paginate(12);
9. return view('core/home', compact('playapp'));
10. }

```

Gambar 5.16 adalah *query* di *controller* yang berguna dalam menghasilkan data untuk ditampilkan pada halaman daftar aplikasi.

```

1. <?php
2. namespace PlayAppSimilar\Model;
3. use Illuminate\Database\Eloquent\Model;
4. class playappsimilar extends Model
5. {
6.     public $timestamps = false; //created_at update
      d_at tidak ada
7. }

```

Gambar 5.17 Model playappsimilar

```

8. Kode pada <?php
9. namespace PlayAppSimilar\Model;
10. use Illuminate\Database\Eloquent\Model;
11. class playappsimilar extends Model
12. {
13.     public $timestamps = false; //created_at update
        d_at tidak ada
14. }

```

Gambar 5.17 adalah model basis data yang menghubungkan basis data dengan *controller*. Proses pembuatannya dengan member perintah di *command line interface* yaitu php artisan make:model playappsimilar.

5.6.2 Implementasi Halaman Detail Aplikasi

Halaman detail aplikasi pada aplikasi ini berperan dalam menampilkan data aplikasi secara lengkap.

```

1. <?php $count = 0; ?>
2.         @foreach(explode(',', $r1-
>screenshots) as $info)
3.             @if($count == 3)
4.                 <?php break; ?>
5.             @else
6.                 <li data-thumb="http:{{$info}};">
7.                 <div class="thumb-
image">  </div>
8.                 </li>
9.                 <?php $count++; ?>
10.            @endif
11.        @endforeach

```

Gambar 5.18 Kode untuk mendapatkan screenshots aplikasi

Kode pada Gambar 5.18 digunakan untuk menampilkan *screenshot* aplikasi yang dibatasi hanya tiga *screenshot* per aplikasi.


```

1. @foreach ($playapp as $r1 )
2.     <?php $gratis = $r1->free; ?>
3.     <div class="col-md-8 single-right">
4.         <h3>{{ $r1->title }}</h3>
5.         <div class="description">
6.             <h5>Biaya @if ($gratis==true)
7.                 <p>Gratis</p>
8.                 @else
9.                 <p>{{ $r1->price }}</p>
10.            @endif
11.        </h5>
12.
13.            <h5><i>Description</i></h5>
14.            <p>{{ $r1->description }}</p>
15.            <div>
16.                <h5>Rating</h5>
17.                <p>{{ $r1->score }}</p>
18. @endforeach

```

Gambar 5.19 Kode untuk menampilkan detail aplikasi

Kode pada Gambar 5.19 digunakan untuk menampilkan detail aplikasi.

```

1. public function detail($id){
2.     $playapp = playappsimilar::join('playapplications
    as pa', 'playappsimilars.appId_play', '=', 'pa.appId')-
    >join('appstoreapplications as asp', 'playappsimilars.appId_app', '=', 'asp.appId')-
    >select('playappsimilars.*', 'asp.url as appurl',
        'pa.title', 'pa.free', 'pa.price', 'pa.description', 'pa.score', 'pa.reviews', 'pa.developerEmail',
        'pa.developerWebsite', 'pa.genre', 'pa.updated', 'pa.developer', 'pa.url as playurl', 'playappsimilars.status', 'pa.screenshots')-
    >where('playappsimilars.id_sim', '=', $id)-
    >get();
3.     return view('core/detail', compact('playapp'));
4. }

```

Gambar 5.20 kueri untuk menampilkan detail aplikasi

Kode pada Gambar 5.20 adalah *query* yang digunakan dalam mengambil detail aplikasi yang berada di tabel *playapplications* dan *appstoreapplications* yang mengacu ke *appId* di tabel *playapps similars*.

5.6.3 Implementasi Halaman Ulasan Aplikasi

Halaman ulasan aplikasi pada aplikasi ini berperan dalam menampilkan ulasan-ulasan pada aplikasi yang tersedia di *play store*. Konten yang disuguhkan dalam menampilkan daftar aplikasi berupa gambar profil pengguna, nama pengguna, tanggal *posting*, penilaian terhadap aplikasi, dan ulasan aplikasi. Untuk melihat ulasan aplikasi, pengguna cukup klik *review* aplikasi yang ada di halaman detail aplikasi.

```

1. public function review($id){
2.     $reviewapp = review::where('app_id','=', $id)-
    >paginate(12);
3.     return view('core/review',compact('reviewapp'
4.     ));
5. }
```

Gambar 5.21 kueri untuk menampilkan ulasan aplikasi

Kode pada Gambar 5.21 adalah *query* yang digunakan dalam mengambil ulasan aplikasi yang berada di tabel *reviews* yang mengacu ke *id* di tabel *playapplications*.

```

1. <?php
2. namespace PlayAppSimilar\Model;
3. use Illuminate\Database\Eloquent\Model;
4. class review extends Model
5. {
6.     public $timestamps = false;
7. }
```

Gambar 5.22 Model reviews

Kode pada Gambar 5.22 adalah model basis data yang menghubungkan basis data dengan *controller*. Proses

pembuatannya dengan member perintah di *command line interface* yaitu php artisan make:model review.

```

1. @extends('layouts.main')
2. @section('content')
3.     <div class="mobiles">
4.         <div class="container">
5.             <div class="w3ls_mobiles_grids">
6.                 <div class="clearfix"> </div>
7.
8.                 @foreach ($reviewapp as $r1)
9.                     <div class="col-md-
10.                        4 agileinfo_new_products_grid agileinfo_new_produ
11.                        cts_grid_mobiles">
12.                         <div class="agile_ecommerce_tab_left
13.                            mobiles_grid">
14.                                <div>
15.                                    
17.                                </div>
18.                                <p>User Pengguna : {{ $r1-
19.                                >userName }}</p>
20.                                <p>Tanggal : {{ $r1->date }}</p>
21.                                <p>Rating : {{ $r1->score }}</p>
22.                                <p>Review : {{ $r1->text }}</p>
23.                            </div>
24.                        </div>
25.                    @endforeach
26.                <div class="clearfix"> </div>
27.            </div>
28.        </div>
29.    <center>
30.        {{ $reviewapp->links() }}
31.    </center>
32. @endsection

```

Gambar 5.23 Kode untuk menampilkan ulasan aplikasi

Kode pada Gambar 5.23 digunakan untuk menampilkan ulasan aplikasi keseluruhan.

5.6.4 Implementasi Halaman Login dan Register

Login dan *register* digunakan sebagai akses developer dalam masuk ke *prototype* aplikasi. Pengguna hanya membutuhkan registrasi dan melakukan *login* dengan memasukkan *email* dan *password* yang telah dimasukkan selama proses registrasi. Untuk membuatnya di laravel menggunakan *command line interface* dengan memberikan perintah `php artisan make:auth` dan `php artisan migrate`. Selanjutnya, *login* dan *register* akan terbuat otomatis dari hak akses maupun akses ke tabel hingga tabel di basis data.

5.6.5 Implementasi Halaman Masukan Data Aplikasi

Halaman untuk memasukkan data aplikasi terdiri dari dua versi yaitu dengan memasukkan `appId` dan memasukkan data secara lengkap. Pengguna hanya memasukkan `appId` aplikasi yang ada di URL play store, lalu menyalin `appId`-nya ke *text field*. Untuk versi kedua jika pengguna tidak mempunyai aplikasi di play store, pengguna cukup memasukkan data secara lengkap di form yang disediakan.

```

1. form class="form-
   horizontal" action="/listings" method="post">
2.     <fieldset>
3.         <div class="form-group">
4.             <label class="col-md-3 control-
               label" for="title">App ID ( Masukkan app id jika
               ada. cth : com.netflix.mediaclient)</label>
5.             <div class="col-md-3">
6.                 <input type="text" name="appid">
7.             </div>
8.         </div>
9.     </fieldset>
10.
11.         <div class="form-group">
12.             <input type="submit" name="submit"
               value="create">
13.         </div>
14.         {{ csrf_field() }}
15.     </form>

```

Gambar 5.24 Kode untuk input form appId

Kode pada Gambar 5.24 adalah formulir *submit* appId yang berguna dalam memasukkan appId ke dalam *controller* untuk selanjutnya diolah.

```
1. public function storeapp(Request $request)
2.     {
3.         $appid = $request->appid;
4.         $data = $request->req_data();
5.         $playapplications = new playapplication;
6.         $playapplications->title = $title;
7.         $playapplications->save();
8.         return view('core/about');
```

Gambar 5.25 Kode untuk memasukkan detail aplikasi dari website

Kode pada gambar 5.25 adalah proses pemanggilan data dari <https://pstoreapi.herokuapp.com/app> yang disimpan di json *encode* dan dilakukan *decode* dan dimasukkan ke variabel yang sesuai dengan nilai yang ada di json tersebut. Lalu disimpan di basis data.

```
1. public function store(Request $request) {
2.     $playapplications = new playapplication;
3.     $playapplications->title = $request-
4.     >title;
5.     $playapplications->save();
6.     return view('core/about');
```

Gambar 5.26 Kode untuk mendapatkan detail aplikasi dari input form

Kode pada Gambar 5.26 adalah proses memasukkan data aplikasi secara lengkap tanpa melakukan pengambilan data seperti yang dilakukan pada Gambar 5.24

```
1. <form class="form horizontal" action ="/listing"  
   method="post">  
2. <div class="col-md-3">  
3.     <input type="text" name="title">  
4. </div>  
5. <div class="col-md-3">  
6.     <select name="genre">  
7.         <option value="ANDROID_WEAR">  
           ANDROID_WEAR</option>  
8. </select>  
9. </div>  
10. </form>
```

Gambar 5.27 kode untuk menampilkan text field detail aplikasi

Kode pada Gambar 5.27 adalah proses memasukkan data secara lengkap tanpa memerlukan appId dan hal yang berkaitan dengan toko aplikasi lainnya. Proses ini diperlukan ketika *developer* belum pernah memasukkan aplikasinya ke berbagai toko aplikasi dan menginginkan aplikasinya dipublikasikan secara luas.

BAB VI

HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai hasil implementasi penelitian yang telah dilaksanakan dan hasil pengujian *prototype* yang dibuat berdasarkan fitur yang ada dalam aplikasi.

6.1 Data aplikasi

Data aplikasi yang diambil dan ditaruh di basis data berhasil menampung semua data yang diperlukan. Sejumlah tujuh tabel menampung data dari *play store* dan *app store* dan satu tabel menampung data hasil pencarian kesamaan aplikasi. Tujuh tabel yang menampung data tersebut yaitu *playapplications*, *comments*, *reviews*, *histograms*, *screenshots*, *permissions*, *appstoreapplications*.

id	title	summary	icon	price	free	mininstalls	maxinstalls	score	reviews	developer	developerEmail
1	Messenger	Messenger — a faster way to message	https://lh5.ggpht.com/DVYA2LR9VhsaF-qgm8u8EwKsCHEY_uk...	0	1	100000000	2147483647	3.90	43843607	Facebook	android-support@fb.com
2	Spotify Music	Spotify gives you instant access to millions of so...	https://lh3.googleusercontent.com/LH17BAZ-XIXGyfwllgZcCC...	0	1	100000000	500000000	4.50	8302298	Spotify Ltd.	android-support@spotify...
3	TextNow - free text + calls	Unlimited FREE texting & calling + free to low-co...	https://lh4.ggpht.com/nHr1-mptolozUSGzxEyZj2R2xG2uq2...	0	1	100000000	500000000	4.40	288829	TextNow, Inc.	textnow.android.feedback...

Gambar 6.1 data di tabel *playapplications*

Pada Gambar 6.1 menggambarkan Tabel *playapplications* terdapat 44154 data aplikasi yang berasal dari 6 koleksi dan 58 kategori.

id	app_id	comments
1	1	Samsung Dex support required, The app is fine. I–...
2	2	Offline songs doesnt play! Please fix! Using lenov...
3	3	Since my 2015 comments, text now seems to work alo...
4	4	Works well, easy to switch between accounts. I hav...
5	5	Deceiving but ok it does everything you9d wan...

Gambar 6.2 data di tabel *comments*

Tabel *comments* terdiri dari 44100 data komentar dari 44154 aplikasi pada Gambar 6.2. Data yang tidak sama disebabkan ada aplikasi yang tidak memiliki *comments*.

id	app_id	userName	userImage	date	url	score	title	text
1	1		https://lh3.googleusercontent.com/-VLpuBv0fhu/AAA...	June 20, 2017	https://play.google.com/store/apps/details?id=com...	5	Is ooo	
2	1		https://lh5.googleusercontent.com/-p8oKa4ru1yo/AAA...	June 20, 2017	https://play.google.com/store/apps/details?id=com...	3	Sending records is so bad	
3	1	lina pointer	https://lh5.googleusercontent.com/-zMR0LEQIU3k/AAA...	June 20, 2017	https://play.google.com/store/apps/details?id=com...	5	I like messenger	is's fast way to get touch ...
4	1	muzahid biyoy	https://lh4.googleusercontent.com/-frig7H51Ci/AAA...	June 20, 2017	https://play.google.com/store/apps/details?id=com...	5	very useful	
5	1	Britney Worley	https://lh3.googleusercontent.com/-Lp2bWMMVNM/AAA...	June 20, 2017	https://play.google.com/store/apps/details?id=com...	1	Not invading my privacy!	
6	1	Gamaliel Rodriguez	https://lh3.googleusercontent.com/-14vqmon054/AAA...	June 20, 2017	https://play.google.com/store/apps/details?id=com...	3	You got to use it soo that's the only bad par...	
7	1	Ace Mark Cabrera	https://lh5.googleusercontent.com/-uF-TX4AIDEM/AAA...	June 20, 2017	https://play.google.com/store/apps/details?id=com...	5	Very good...	

Gambar 6.3 data di tabel *reviews*

Tabel *reviews* terdiri dari 473544 data ulasan dari 44154 aplikasi pada Gambar 6.3.

id	app_id	value
1	1	6330455, 2022365, 4458548, 6895407, 25916245
2	2	348490, 158502, 399572, 1315421, 6512147
3	3	19324, 9049, 21497, 41345, 213849
4	4	171303, 78398, 165304, 387506, 698340
5	5	258959, 159643, 368507, 835058, 3407213

Gambar 6.4 data di tabel *histograms*

Tabel *histograms* terdiri dari 44154 penilaian skala dari 44154 aplikasi pada Gambar 6.4.

id	app_id	url
1	1	//lh3.googleusercontent.com/NAZ5jkXAU2bL7luKqUwbpl...
2	2	//lh3.googleusercontent.com/0A2Xd5l_bHkvyzYva1e7ZY...
3	3	//lh3.googleusercontent.com/l5zKkOKa9mpCexj_TUI-7u...
4	4	//lh3.googleusercontent.com/cl1FPJ4v8VWxd2qkMk7rYG...
5	5	//lh3.googleusercontent.com/dar060xShkqnJjWC2j_Eaz...

Gambar 6.5 data di tabel *screenshots*

Tabel *screenshots* terdiri dari 44097 penilaian skala dari 44154 aplikasi pada Gambar 6.5. Nilai yang tidak sama dikarenakan ada aplikasi yang tidak memiliki *screenshots*.

id	app_id	permission	description
1	1	find accounts on the device	Allows the app to get the list of accounts known b...
2	1	add or remove accounts	Allows the app to perform operations like adding a...
3	1	read your own contact card	Allows the app to read personal profile informatio...
4	1	read your contacts	Allows the app to read data about your contacts st...
5	1	modify your contacts	Allows the app to modify the data about your conta...

Gambar 6.6 data di tabel *permissions*

Tabel *permissions* terdiri dari 680927 data perizinan aplikasi dari 44154 aplikasi pada Gambar 6.6.

id_app	id_xc	appid	name	url	description	icon
1	1196233335	com.science-inc.Yarn	Yarn - Chat Fiction	https://itunes.apple.com/us/app/yarn-chat-fiction/...	Yarn - Text Stories Ever wanted to snog through ...	http://s3.mzstatic.com/image/thumb/Purple111/v4/8...
2	1024816709	tv.telepathic.hooked	HOOKED - Chat Stories	https://itunes.apple.com/us/app/hooked-chat-stories/...	HOOKED lets you read amazing chat stories FREE. Th...	http://s3.mzstatic.com/image/thumb/Purple111/v4/3...
3	379693831	com.audible.iphone	Audible - audio books, original series & podcasts	https://itunes.apple.com/us/app/audible-audio-book/...	Welcome to Audible. We're an Amazon company, and ...	http://s3.mzstatic.com/image/thumb/Purple122/v4/9...
4	302554613	com.amazon.Lassen	Kindle - Read eBooks, Magazines & Textbooks	https://itunes.apple.com/us/app/kindle-read-ebooks/...	Turn your iPhone or iPad into a Kindle with the fr...	http://s3.mzstatic.com/image/thumb/Purple122/v4/d...
5	306310789	com.fivemobile.watpad	Wattpad - Read unlimited books and eBooks	https://itunes.apple.com/us/app/wattpad-read-unlim/...	Discover Wattpad. At Wattpad, we're connecting a	http://s4.mzstatic.com/image/thumb/Purple111/v4/8...

Gambar 6.7 data dari tabel appstoreapplications

Tabel appstoreapplications terdiri dari 18336 data aplikasi yang terhimpun dari 13 koleksi dan dari masing-masing koleksi itu terdapat 70 kategori pada Gambar 6.7.

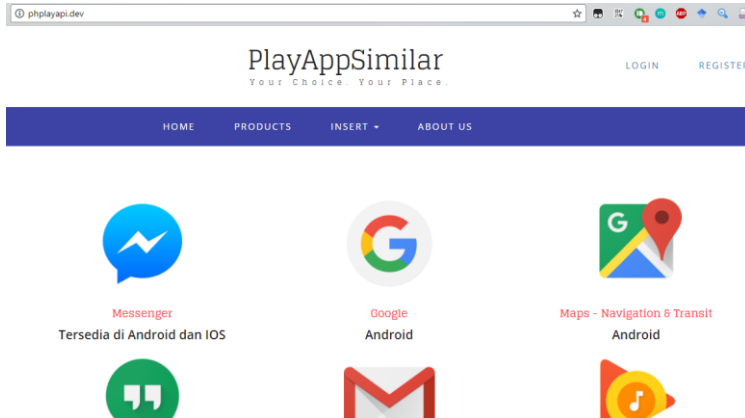
6.2 Hasil Prototipe Aplikasi

Hasil prototipe aplikasi berupa antar muka pengguna yang siap dipakai dengan kapabilitas terbatas hanya pada menampilkan daftar aplikasi, melihat detail produk yang dipilih oleh pengguna, memasukkan data aplikasi yang dibuat oleh *developer* serta login dan register untuk *user*.

Terdapat enam tampilan antar muka antara lain halaman daftar aplikasi, halaman detail aplikasi, halaman ulasan aplikasi, halaman masukan data aplikasi berdasarkan appId, Halaman masukan data secara rinci, halaman *login*, dan alaman registrasi. Berikut adalah penjelasan mengenai masing-masing fitur halaman.

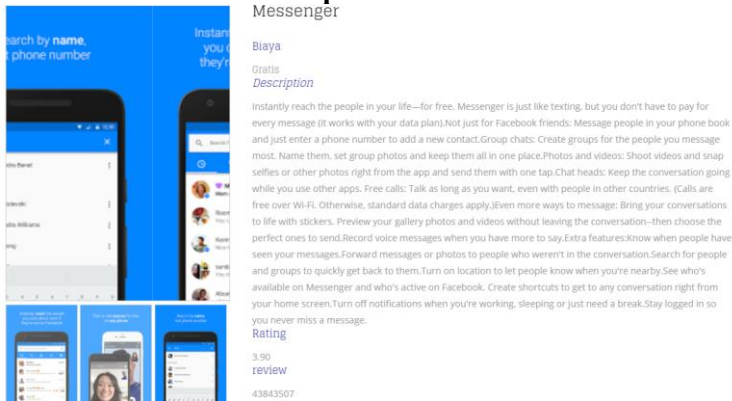
6.2.1 Halaman Daftar Aplikasi

Halaman pada Gambar 6.8 adalah tampilan awal dari playappsimilar, nama *prototype* aplikasi untuk menampilkan daftar aplikasi dari hasil *scraping* yang berhasil diambil. Pengguna bisa mengetahui aplikasi tersebut tersedia di *play store* dan *app store* maupun *play store* saja atau sebaliknya berasal dari indikator yang ada dibawah judul.



Gambar 6.8 Halaman Daftar Aplikasi

6.2.2 Halaman Detail Aplikasi

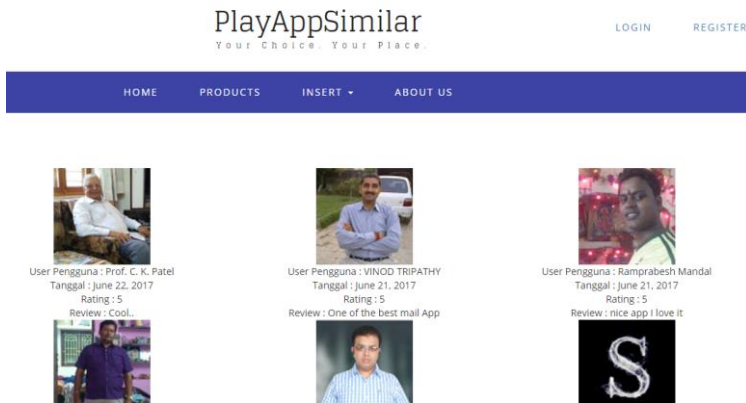


Gambar 6.9 Halaman Detail Aplikasi

Halaman pada Gambar 6.9 adalah tampilan untuk menampilkan informasi aplikasi secara rinci. Pengguna bisa mengetahui informasi aplikasi secara lengkap dan disediakan *screenshot* aplikasi dan *hyperlink* yang menuju toko aplikasi yang tersedia.

6.2.3 Halaman Ulasan Aplikasi

Halaman pada Gambar 6.10 adalah tampilan untuk menampilkan ulasan aplikasi dari aplikasi yang dipilih. Pengguna bisa mengetahui berbagai ulasan aplikasi secara lengkap.



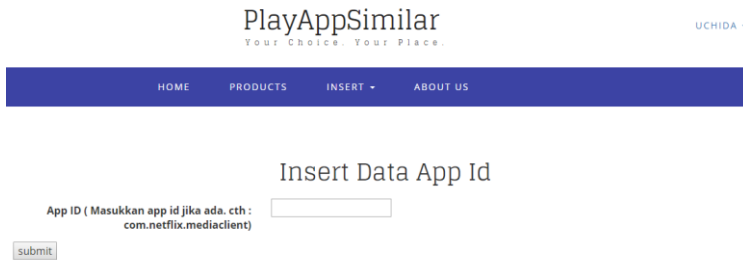
Gambar 6.10 Halaman Ulasan Aplikasi

6.2.4 Halaman Masukan Data Secara Rinci

Gambar 6.11 Halaman masukan data secara rinci

Halaman pada Gambar 6.11 adalah tampilan untuk memasukkan data aplikasi secara rinci. Pengguna yang belum pernah memasukkan karyanya ke *play store* dan *app store* dapat memasukkan aplikasi yang dibuat ke formulir di halaman ini dengan mengisi data-data yang ada di formulir tersebut.

6.2.5 Halaman Masukan Data Berdasarkan Appid

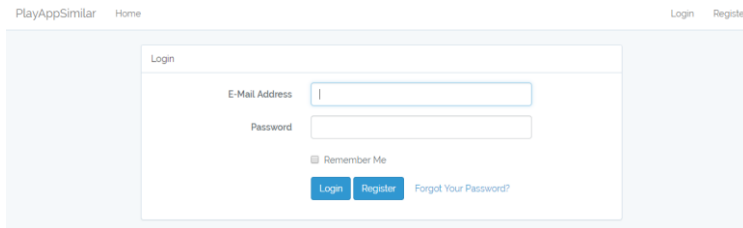


The screenshot shows the PlayAppSimilar website header with the logo and tagline 'Your Choice. Your Place.' and a navigation bar with links: HOME, PRODUCTS, INSERT, and ABOUT US. The main content area is titled 'Insert Data App Id'. Below the title, there is a text input field for 'App ID (Masukkan app id jika ada. cth : com.netflix.mediaclient)'. A 'submit' button is located below the input field.

Gambar 6.12 Halaman masukan data berdasarkan appid

Halaman pada Gambar 6.12 adalah tampilan untuk memasukkan appId aplikasi yang tersedia di *play store* maupun *app store*. Pengguna yang mempunyai karya berupa aplikasi dan sudah diunggah ke *play store* atau *app store* dapat memasukkan appId aplikasi dan sistem secara otomatis akan melakukan *scraping* ke <http://pstoreapi.herokuapp.com/app> dan <http://astoreapi.herokuapp.com/app> sesuai appId.

6.2.6 Halaman *login*



The screenshot shows the PlayAppSimilar website header with the logo and tagline 'Your Choice. Your Place.' and a navigation bar with links: HOME, Login, and Register. The main content area is titled 'Login'. Below the title, there is a form with two input fields: 'E-Mail Address' and 'Password'. Below the input fields, there is a checkbox labeled 'Remember Me'. At the bottom of the form, there are three buttons: 'Login', 'Register', and 'Forgot Your Password?'.

Gambar 6.13 Halaman *login*

Halaman pada Gambar 6.13 adalah tampilan untuk masuk akun ke *prototype* aplikasi yang didalamnya digunakan untuk memasukkan data aplikasi. Pengguna memasukkan email address yang dimasukkan saat registrasi dan *password*.

6.2.7 Halaman Registrasi

The image shows a web registration form. At the top, there is a header with 'PlayAppSimilar' and 'Home' on the left, and 'Login' and 'Register' on the right. The main content area is titled 'Register' and contains four input fields: 'Name', 'E-Mail Address', 'Password', and 'Confirm Password'. Below these fields is a blue button labeled 'Register'.

Gambar 6.14 Halaman registrasi

Halaman pada Gambar 6.14 adalah tampilan untuk registrasi akun untuk masuk ke dalam *prototype* aplikasi. Pengguna dapat mengisi formulir yang disediakan, yaitu nama, alamat *email*, *password*, dan konfirmasi *password*.

6.3 Hasil Analisis perbandingan

Pada bab ini akan dijelaskan analisis perbandingan fungsi levenshtein dengan penggabungan fungsi antara levenshtein dan *similar text*, perbandingan ini dilakukan untuk mengetahui tingkat kesalahan yang dilakukan dalam mencari kesamaan data.

6.3.1 Perbandingan Antara Levenshtein Dengan Gabungan Levenshtein dan Similar Text

Dalam menentukan perbandingan, mula-mula ditentukan jumlah sampel data yang diperlukan. Cara menentukannya menggunakan metode slovin dengan toleransi kesalahan 5% dan tingkat presisi 95%. Dalam data ini menggunakan jumlah populasi 2635 data.

$$n = \frac{2635}{1 + 2635 \cdot (0.05)^2} \quad (6.1)$$

Nilai n didapatkan hasil 347.2 yang dibulatkan menjadi 347 data. Dalam hal ini digunakan 347 sampel data untuk mencakup 2635 data.

Tahapan setelah itu mencari nilai kesalahan di levenshtein dan didapatkan hasil 32 kesalahan, maka untuk mencari nilai keakuratan didapatkan hasil 90.7%.

$$n = \frac{347 - 32}{347} \times 100\% \quad (6.2)$$

Lalu untuk mencari nilai kesalahan di gabungan fungsi antara levenshtein dan *similar text* didapatkan hasil 19 kesalahan, maka untuk mencari nilai keakuratan didapatkan hasil 94.5%

$$n = \frac{347 - 19}{347} \times 100\% \quad (6.3)$$

Dari perhitungan tersebut, gabungan fungsi levenshtein dan fungsi *similar text* lebih baik dengan tingkat akurasi 94.5 %.

6.4 Usability Testing

Pada pengembangan aplikasi, untuk mendapatkan banyak masukan, aplikasi diuji dengan membuat *Minimum Viable Product* (MVP) dari seluruh *use case* yang ada. *Usability testing* dilakukan dengan satu iterasi uji coba. Seluruh hasil dari *usability testing* yang dilakukan didokumentasikan pada dokumen protokol dan hasil uji coba.

Pada akhir setiap sesi *usability testing* yang dilakukan, responden diminta untuk mengisi kuesioner *System Usability Scale* (SUS) yang digunakan untuk mengukur kemudahan dari penggunaan aplikasi. Skenario *Test Case* dapat dilihat pada lampiran C

Tabel 6. 1 Daftar Test Case

ID Test Case	Test Case
TC01	Memilih aplikasi
TC02	Melihat detail aplikasi
TC03	Daftar akun developer
TC04	Login ke halaman developer
TC05	Input data aplikasi
TC06	Posting aplikasi

Hasil dari kuesioner *System Usability Scale* menunjukkan aplikasi yang dikembangkan memiliki kemudahan dalam

penggunaan sebesar 74.5% seperti yang terlihat pada Lampiran B. Pada kuisioner tersebut menunjukkan bahwa banyak pengguna yang setuju dengan pernyataan yang diajukan kepada mereka.

Tabel 6. 2 Hasil Justifikasi System Usability Scale

Pernyataan	Justifikasi
Pengguna berpikir bahwa saya ingin menggunakan system ini sesering mungkin	Setuju
Pengguna berpikir sistem ini sangat rumit dan tidak perlu	Tidak Setuju
Pengguna berpikir sistem ini mudah digunakan	Setuju
Pengguna berpikir saya akan membutuhkan bantuan seseorang untuk dapat menggunakan sistem ini	Tidak Setuju
Pengguna menemukan seluruh fungsi yang telah terintegrasi dengan sangat baik	Setuju
Pengguna berpikir terlalu banyak inkonsistensi dari sistem ini	Tidak Setuju
Pengguna berpikir banyak orang yang akan dapat belajar menggunakan sistem ini dengan cepat	Setuju
Pengguna berpikir bahwa sistem ini sangat sulit untuk digunakan	Tidak Setuju
Pengguna merasa sangat percaya diri dalam menggunakan sistem ini	Setuju
Pengguna perlu belajar banyak hal sebelum dapat menggunakan sistem ini	Tidak Setuju

Adapun hasil dari kuisioner diatas dapat dilihat pada lampiran B, untuk lembar persetujuan dapat dilihat pada lampiran E dan lampiran F untuk kuisioner mengenai *System Usability Scale* (SUS).

(Halaman ini sengaja dikosongkan)

BAB VII

KESIMPULAN DAN SARAN

Pada bab ini dijelaskan mengenai kesimpulan dari hasil penelitian dan saran untuk pengembangan penelitian di kemudian hari.

7.1 Kesimpulan

Berdasarkan dengan pengerjaan tugas akhir dengan judul “rancang bangun agregasi toko aplikasi mobile berbasis web untuk meningkatkan pendayagunaan aplikasi multi platform” yang telah dilakukan dapat disimpulkan:

1. Proses pengerjaan dimulai dari mengidentifikasi website API. Lalu, pembuatan *entity relationship diagram* (ERD) sehingga membentuk basis data yang efisien dan tidak redundan. Setelah itu pengambilan data menggunakan cURL yang pengisiannya sesuai tabel-tabel yang telah dibuat meliputi playapplication, permissions, comments, histograms, screenshots, reviews, dan appstoreapplications. Lalu proses terakhir adalah pembuatan alur kerja *website* direktori aplikasi menggunakan *use case*.
2. Hasil pengambilan data dari *play store* didapatkan 44100 aplikasi dan dari *app store* didapatkan 18318 aplikasi. Hasil pengambilan data itu dicari kesamaan aplikasi dan didapatkan 761 aplikasi dari total 2635 aplikasi yang dicari kesamaannya menggunakan fungsi levenshtein. Sedangkan setelah dilakukan penggabungan fungsi levenshtein dan fungsi similar text, didapatkan kesamaan 1696 aplikasi dari total 2635. Sehingga, metode penggabungan dua fungsi diimplementasikan ke aplikasi karena hasilnya yang lebih baik.
3. Hasil analisis perbandingan yang telah dijabarkan pada bab 6 diketahui bahwa terdapat perbedaan hasil *similarity* pada kedua metode. Hal ini terjadi karena fungsi levenshtein yang mencapai keakuratan 90.7% tidak cukup mencapai titik keakuratan yang dibutuhkan sedangkan penggabungan antara fungsi levenshtein dan fungsi similar text yang

mencapai keakuratan 94.5% sudah mencapai tingkat keakuratan dalam mencari persamaan aplikasi di *play store* dan *app store*.

4. Hasil dari *usability testing* yang dilakukan ke sebanyak 10 responden menunjukkan aplikasi yang dikembangkan memiliki kemudahan dalam penggunaan sebesar 74.5%. Angka ini tidak kecil tetapi tidak juga besar, sehingga bisa dilakukan pengembangan lebih lanjut untuk mendapatkan *usability testing* yang lebih besar.

7.2 Saran

Berdasarkan penelitian pada tugas akhir ini dapat diberikan saran ataupun masukan untuk pengembangan selanjutnya, antara lain:

1. Penggunaan fungsi levenshtein kurang cocok dalam mencari kesamaan data aplikasi yang prosesnya membandingkan per huruf, disarankan untuk membandingkan dengan blok-blok kata di appId aplikasi.
2. Prototipe ini sebaiknya lebih ke arah memperkaya metadata aplikasi secara otomatis daripada menambahkan fitur memasukkan data manual untuk pengembang aplikasi.
3. User Interface pada prototipe ini dapat dikembangkan lebih lanjut untuk meningkatkan kepuasan pengguna dalam menggunakan prototipe ini.

DAFTAR PUSTAKA

- [1] A. Firat, S. Madnick, N. A. Yahaya, C. W. Kuan, dan S. Bressan, "Information Aggregation using the Caméléon# Web Wrapper," no. August, 2005.
- [2] R. H. Walsh dan T. A. Musgrove, "Content aggregation method and apparatus for an on-line product catalog," vol. 2, no. 12, 2006.
- [3] facundoolano, "google-play-scraper." [Daring]. Tersedia pada: <https://github.com/facundoolano/google-play-scraper>. [Diakses: 23-Feb-2017].
- [4] facundoolano, "app-store-scraper." [Daring]. Tersedia pada: <https://github.com/facundoolano/app-store-scraper>. [Diakses: 23-Feb-2017].
- [5] "Laravel - The PHP Framework For Web Artisans." [Daring]. Tersedia pada: <https://github.com/laravel/laravel>. [Diakses: 24-Feb-2017].
- [6] T. Nugraha, "Tutorial Dasar Laravel," hal. 1–43, 2014.
- [7] J. NIELSEN, "Usability 101:Introduction to usability." hal. 1–13, 2012.
- [8] F. Z. Ridwan, D. Hardianto, dan Y. G. Sucahyo, "Analisa Usability Untuk Mengetahui User Experience Pada Aplikasi Berbasis Web," *Konf. Nas. Sist. dan Inform.*, hal. 186–191, 2008.
- [9] Michael Gilleland dan Merriam Park Software, "Levenshtein Distance." [Daring]. Tersedia pada: [https://people.cs.pitt.edu/~kirk/cs1501/Pruhs/Spring2006/assignments/editdistance/Levenshtein Distance.htm](https://people.cs.pitt.edu/~kirk/cs1501/Pruhs/Spring2006/assignments/editdistance/Levenshtein%20Distance.htm). [Diakses: 16-Jun-2017].
- [10] Boyke Dian Triwahyudhi, "Web Scraping, Trend ancaman Bisnis dan Teknologi Selanjutnya." [Daring]. Tersedia pada: <https://indosystem.com/blog/web-scraping-trend-ancaman-bisnis-dan-teknologi-berikutnya/>. [Diakses: 24-Feb-2017].
- [11] NATANAEL YABES WIRAWAN, "RANCANG BANGUN EKSTRAKSI TOPIK FITUR PRODUK DARI ULASAN PENGGUNA ONLINE DENGAN

LATENT DIRICHLET ALLOCATION,” 2016.

- [12] T. Connolly dan C. Begg, “Database Systems: A Practical Approach to Design, Implementation, and Management,” hal. 1–1425, 2005.

BIODATA PENULIS



Penulis lahir di Bekasi pada tanggal 2 Desember 1995. Penulis merupakan anak ketiga dari tiga bersaudara. Penulis telah menempuh pendidikan formal di sekolah negeri mulai dari SDN Harapan Jaya XV hingga lulus pada tahun 2007, SMPN 5 Bekasi hingga lulus pada tahun 2010, dan SMAN 4 Bekasi hingga lulus pada tahun 2013. Setelah lulus, penulis melanjutkan ke jenjang perguruan tinggi negeri di Surabaya, yakni Jurusan Sistem Informasi Institut Teknologi

Sepuluh Nopember Surabaya. Sebagai mahasiswa penulis aktif dalam urusan akademik, non akademik maupun organisasi. Tercatat penulis pernah menjadi staff pada Departemen Riset dan Teknologi di Himpunan Mahasiswa Sistem Informasi (HMSI) ITS Surabaya, staff Syiar (Media Informasi) pada Kajian Islam Sistem Informasi (KISI) ITS, dan staff media pada Jemaah Masjid Manarul Ilmi (JMMI) ITS. Selain organisasi formal, penulis juga pernah mengikuti organisasi non-formal, yakni Unit Kegiatan Mahasiswa pada ITS Robotika menjadi staff Riset dan Teknologi dan kepala divisi Komunikasi dan Informasi. Selain organisasi, penulis juga aktif dalam kepanitiaan, baik panitia dalam organisasi yang diikutinya, maupun di luar organisasi. Penulis juga pernah menjalani kerja praktik di PT. Pelabuhan Tanjung Priok di Jakarta selama 2 bulan pada tahun 2016.

Untuk mendapatkan gelar Sarjana Komputer (S.Kom), penulis mengambil laboratorium bidang minat Sistem Enterprise (SE). Untuk kepentingan penelitian penulis juga dapat dihubungi melalui e-mail: prireзки@gmail.com.

(Halaman ini sengaja dikosongkan)

Lampiran A

Hasil Translasi Wawancara

Narasumber : Dwi Karya Maha Putera

ID	Question and Answer
QA1	<p>Apakah karya anda sudah pernah di unggah ke toko aplikasi ?</p> <p>Sudah Pernah.</p>
QA2	<p>Kira-kira sudah ada berapa yang pernah di unggah?</p> <p>Sejauh ini ada 3 aplikasi yang sudah pernah di unggah.</p>
QA3	<p>Dari 3 aplikasi itu ada yang belum pernah diunggah?</p> <p>Ada 1 aplikasi karena Nama aplikasinya menyebut nama aplikasi yang sudah ada dan sudah dipatenkan sehingga melanggar hak cipta.</p>
QA4	<p>Dari 3 aplikasi itu, apakah ada rencana untuk mengunggah ke toko aplikasi pihak ketiga?</p> <p>Kalau saya sih untuk aplikasi itu pusatnya di play store, lalu sisanya bakal ada yang mengunggah sendiri aplikasi di play store ke toko aplikasi lain.</p>
QA5	<p>Dari 3 aplikasi itu, apakah ada rencana untuk mengunggah ke toko aplikasi pihak ketiga?</p> <p>Kalau saya sih untuk aplikasi itu pusatnya di play store, lalu sisanya bakal ada yang mengunggah sendiri aplikasi di play store ke toko aplikasi lain.</p>
QA6	<p>Apakah ada rencana untuk membuat aplikasi untuk versi app store?</p> <p>Pasti ada, tapi dengan melihat statistic terlebih dahulu yang ada di Indonesia, ketika pengguna menginginkan hal tersebut maka baru dibuat untuk versi app storenya.</p>
QA7	<p>Apakah ada kesulitan dalam mempublikasikan karyanya selain pelanggaran hak cipta ?</p> <p>mengenai penjelasan apa saja yang tidak boleh dilanggar, dan di play store ini semuanya ditaruh di policy and</p>

	agreement dimana kebanyakan orang tidak akan melihat hal tersebut. Alangkah lebih baik jika diberikan contoh-contoh penerapannya.
--	---

Narasumber : Sigit Prabowo

ID	Question and Answer
QA1	Apakah karya anda sudah pernah di unggah ke toko aplikasi ? Sudah Pernah.
QA2	Kira-kira sudah ada berapa yang pernah di unggah? Sejauh ini ada 7 aplikasi yang sudah pernah di unggah.
QA3	Dari 7 aplikasi itu ada yang belum pernah di unggah? Sampai saat ini belum ada karena masih dalam proses pengembangan.
QA4	Dari 7 aplikasi itu, apakah ada rencana untuk mengunggah ke toko aplikasi pihak ketiga? Kalau saya sih boleh saja asalkan calon pengguna saya banyak yang mengakses ke sana, tapi sampai saat ini belum ada informasi yang mengatakan hal demikian.
QA5	semisal ada toko aplikasi pihak ketiga yang proses unggah aplikasinya itu mudah namun belum ada calon pengguna yang mengetahui toko aplikasi tersebut, apakah anda mau mengunggah ke toko aplikasi tersebut? Kalau menurut saya sih lebih mentingin calon pengguna jadi untuk saat ini akan terus unggah ke play store.
QA6	Apakah ada kesulitan dalam mempublikasikan karyanya? Ada, pembayarannya harus via credit card saat beli akun google play. Lalu, review buat aplikasi saat mau di publikasi itu memerlukan waktu 5 jam, itu berlaku juga saat melakukan update aplikasi.

Lampiran B

Hasil Perhitungan System Usability Scale (SUS)

SUS Calculation												
Peserta	Jenis Kelamin	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Skor SUS
M. Fahmi Zamroni	Pria	4	2	5	2	4	1	4	2	5	2	82,5%
Ashma Hanifah	Wanita	4	3	4	2	4	3	4	2	3	2	67,5%
Stezar Priansya	Pria	4	2	4	2	2	4	3	2	4	2	62,5%
Pramita Lucianna	Wanita	5	1	5	2	5	1	5	2	4	5	82,5%
Delina Rahayu Effendi	Wanita	4	3	4	2	3	2	3	2	4	2	67,5%
Umar al-aqsho	Pria	4	2	4	2	2	2	4	2	2	4	60,0%
Ervi Ritya Zulfima	Wanita	4	2	5	1	4	1	5	1	5	1	92,5%
Tetha Valianta	Pria	4	2	4	4	4	2	4	2	4	2	70,0%
Marina Safitri	Wanita	4	2	5	2	5	2	4	1	4	3	80,0%
Fitri Larasati	Wanita	4	2	4	2	4	2	4	1	5	2	80,0%
Total Usabilitas												74,5%

B - 2 -

(Halaman ini sengaja dikosongkan)

C - 2 -

```

1 <!doctype html>
2 <?php
3 <title>PHP CURL Google Play API</title>
4 <style>
5 body{
6     margin: 100px 300px 0 300px;
7     font-family: sans-serif;
8 }
9
10 .panel{
11     margin-bottom: 30px;
12     border: 1px solid #eaebeb;
13     padding: 30px;
14 }
15 </style>
16 </head>
17 <body>
18
19
20 <?php
21 include('koneksi.php');
22 $query = "select appid from applications";
23 $result = $dbh->query($query)->fetchAll();
24 foreach($result as $r){
25     $a = $r['appid'];
26
27 $service_url = "https://googleplayapi.herokuapp.com/app/$a";
28 curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
29 curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 0);
30 curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);
31 //execute the session
32 $curl_response = curl_exec($curl);
33 //finish off the session
34 curl_close($curl);
35 $sval = json_decode($curl_response, true);
36
37 $minInstalls = htmlspecialchars($sval['minInstalls'], ENT_QUOTES);
38
39 }
40 $minInstalls = "";
41
42 if(isset($sval['maxInstalls'])){
43     $maxInstalls = htmlspecialchars($sval['maxInstalls'], ENT_QUOTES);
44 }
45 else{
46     $maxInstalls = "";
47 }
48
49 if(isset($sval['reviews'])){
50     $reviews = htmlspecialchars($sval['reviews'], ENT_QUOTES);
51 }
52 else{
53     $reviews = "";
54 }
55
56 }
57
58 }
59
60 }
61
62 }
63
64 }
65
66 }
67
68 }
69
70 }
71
72 }
73
74 }
75
76 }
77
78 }
79
80 }
81
82 }
83
84 }
85
86 }
87
88 }
89
90 }
91
92 }
93
94 }
95
96 }
97
98 }
99
100 }
101
102 }
103
104 }
105
106 }
107
108 }
109
110 }
111
112 }
113
114 }
115
116 }
117
118 }
119
120 }
121
122 }
123
124 }
125
126 }
127
128 }
129
130 }
131
132 }
133
134 }
135
136 }
137
138 }
139
140 }
141
142 }
143
144 }
145
146 }
147
148 }
149
150 }
151
152 }
153
154 }
155
156 }
157
158 }
159
160 }
161
162 }
163 </body>

```

Gambar C-2 Pengambilan satu data aplikasi google play secara utuh

```

20 <?php
21 include('koneksi_ip.php');
22 $query = "select category from kategori";
23 $result1 = $dbh->query($query)->fetchAll();
24 foreach($result1 as $s1){
25     $s = $s1['category'];
26     $service_url = "https://applestoreapi.herokuapp.com/list/new_ios/$s/us/100";
27     $curl = curl_init($service_url);
28     curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
29     curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 0);
30     curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);
31     //execute the session
32     $curl_response = curl_exec($curl);
33     //finish off the session
34     curl_close($curl);
35     $curl_json = json_decode($curl_response, true);
36     //myfile = fopen("scrap.txt", "w") or die("Unable to open file!");
37     foreach ($curl_json as $val) {
38         # code...
39         //echo("<div class='panel'> <img width='50' src='https://".$val['title']".
40         if(isset($val['id'])){
41             $id = htmlspecialchars($val['id'], ENT_QUOTES);
42         }
43         else{
44             $id = "";
45         }
46         if(isset($val['appId'])){
47             $appId = htmlspecialchars($val['appId'], ENT_QUOTES);
48         }
49         else{
50             $appId = "";
51         }
52         if(isset($val['title'])){
53             $title = htmlspecialchars($val['title'], ENT_QUOTES);
54         }
55         else{
56             $title = "";
57         }
58         if(isset($val['icon'])){
59             $icon = htmlspecialchars($val['icon'], ENT_QUOTES);
60         }
61         else{
62             $icon = "";
63         }
64         if(isset($val['url'])){
65             $url = htmlspecialchars($val['url'], ENT_QUOTES);
66         }
67         else{
68             $url = "";
69         }
70         if(isset($val['price'])){
71             $price = htmlspecialchars($val['price'], ENT_QUOTES);
72         }
73         else{
74             $price = "";
75         }
76     }
77     $currency = htmlspecialchars($val['currency'], ENT_QUOTES);
78     }
79     else{
80         $currency = "";
81     }
82     if(isset($val['free'])){
83         $free = htmlspecialchars($val['free'], ENT_QUOTES);
84     }
85     else{
86         $free = "";
87     }
88     if(isset($val['developer'])){
89         $developer = htmlspecialchars($val['developer'], ENT_QUOTES);
90     }
91     else{
92         $developer = "";
93     }
94     if(isset($val['developerUrl'])){
95         $developerUrl = htmlspecialchars($val['developerUrl'], ENT_QUOTES);
96     }
97     else{
98         $developerUrl = "";
99     }
100     if(isset($val['developerId'])){
101         $developerId = htmlspecialchars($val['developerId'], ENT_QUOTES);
102     }
103     else{
104         $developerId = "";
105     }
106     if(isset($val['description'])){
107         $description = htmlspecialchars($val['description'], ENT_QUOTES);
108     }
109     else{
110         $description = "";
111     }
112     }
113     if(isset($val['released'])){
114         $released = htmlspecialchars($val['released'], ENT_QUOTES);
115     }
116     else{
117         $released = "";
118     }
119     }
120     $query1 = "select appId from applications where appId='".$appId."'";
121     $result2 = $dbh->query($query1)->fetch();
122     if($result2){
123         $table = "applications"; //ubah ke nama tabel
124         $fields = "id, appId, title, icon, url, price, currency, free, description, developer, developerUrl, developerId, released"; //kolomnya, kalo > 2 pindahkan pakai koma
125         $sql = "SELECT * FROM $table WHERE appId='".$appId."'"; // untuk semua jumlah kolomnya, ganti koma
126         $array = array($id, $appId, $title, $icon, $url, $price, $currency, $free, $description, $developer, $developerUrl, $developerId, $released); // sesuai jumlah kolom juga
127         $dbh->prepare("INSERT INTO $table ($fields) VALUES ($sql)");
128         if($dbh->execute($array)){
129             echo "sukses";
130         } else {
131             echo "gagal";
132         }
133     }

```

Gambar C-3 Pengambilan data app store bagian daftar aplikasi

Gambar C-4 Pengambilan satu data aplikasi app store secara utuh

```

160 $ports = @($screenshots, $letscreenshots, $appleletscreenshots, $supportfeatures, $supportfeatures MMCX_0_A1, $10_00 And apple = 'apple';
161
162 $f($idh)=wex($query2)
163 {
164     echo 'subex';
165 }
166
167 $v = else {
168     echo 'gaga';
169 }
170
171

```

Lampiran D

Kode Pembuatan Aplikasi

```
namespace PlayAppSimilar\Http\Controllers;

use Illuminate\Http\Request;
use PlayAppSimilar\Model\playappsimilar;

class playappsimilarController extends Controller
{
    protected $layout = 'layouts.main';

    public function index(){
        $playapp = playappsimilar::join('playapplications', 'playappsimilar.appid_play', '=',
        'playapplications.appid')->join('appstoreapplications', 'playappsimilar.appid_app', '=',
        'appstoreapplications.appid')->select('playappsimilar.*', 'appstoreapplications.*',
        'playapplications.*')->orderBy('playapplications.maxInstalls', 'desc')->paginate(12);
        return view('core/home', compact('playapp'));
    }

    public function detail($id){
        $playapp = playappsimilar::join('playapplications as pa', 'playappsimilar.appid_play', '=', 'pa.appid')->join('appstoreapplications
        as asp', 'playappsimilar.appid_app', '=', 'asp.appid')->select('playappsimilar.*', 'asp.url as appurl', 'pa.title', 'pa.free',
        'pa.price', 'pa.description', 'pa.score', 'pa.reviews', 'pa.developeremail', 'pa.developerwebsite', 'pa.genre', 'pa.updated'
        , 'pa.developer', 'pa.url as playurl', 'playappsimilar.status', 'pa.screenshots')->where('playappsimilar.id_sim', '=', $id)->get();
        return view('core/detail', compact('playapp'));
    }

    public function detail_app($id){
        $appstores = playappsimilar::join('appstoreapplications', 'playappsimilar.appid_app', '=',
        'appstoreapplications.appid')->select('playappsimilar.*', 'appstoreapplications.*')->where('appstoreapplications.id_app', '=',
        $id)->get();
        return view('lists', compact('appstores'));
    }

    public function about(){
        return view('core/about');
    }
}
```

Gambar D-1 Controller playappsimilar mengatur daftar aplikasi dan detail aplikasi

```
1  @extends('layouts.main')
2
3  @section('content')
4      <div class="mobiles">
5          <div class="container">
6              <div class="wals_mobiles_grids">
7                  <div class="clearfix"> </div>
8                  @foreach ($playapp as $r1)
9                      <div class="col-md-4 agileinfo_new_products_grid agileinfo_new_products_grid_mobiles">
10                         <div class="agile_ecommerce_tab_left mobiles_grid">
11                             <div>
12                                 
13                                 <div>
14                                     <h5><a href="list/{{($r1->id_sim)}}">{{($r1->title_play)}</a></h5>
15                                     <div class="simplecart_shelfitem">
16                                         <php $a = $r1->status; ?>
17                                         @if ($a=="1")
18                                             <p>Tersedia di <i>Android</i> dan <i class="item_price">IOS</i></p>
19                                         @else
20                                             <p><i class="item_price">Android</i></p>
21                                         @endif
22                                     </div>
23                                 </div>
24                             </div>
25                             @endforeach
26                         <div class="clearfix"> </div>
27                     </div>
28                 </div>
29             </div>
30             <center>
31                 {{ $playapp->links() }}
32             </center>
33         @endsection
```

Gambar D-2 Menampilkan daftar aplikasi

[illegible]

Gambar D-3 Menampilkan detail aplikasi


```

<extends('layouts.main')
</section('content')] ]]
<div class="about">
  <div class="container">
    <center>
      <h3>Insert Data Aplikasi</h3>
      <p>
        Developer dapat memasukkan data aplikasi yang karyanya belum dipublikasikan di play store atau app store.
      </p>
    </center>
  </div>
  <div class="col-md-6">
    <form class="form-horizontal" action="/" listing method="post">
      <fieldset>
        <div class="form-group">
          <label class="col-md-3 control-label" for="title">title</label>
          <div class="col-md-3">
            <input type="text" name="title">
          </div>
        </div>
      </fieldset>
      <fieldset>
        <div class="form-group">
          <label class="col-md-3 control-label" for="title">summary</label>
          <div class="col-md-3">
            <input type="text" name="summary">
          </div>
        </div>
      </fieldset>
      <fieldset>
        <div class="form-group">
          <label class="col-md-3 control-label" for="title">icon</label>
          <div class="col-md-3">
            <input type="text" name="icon">
          </div>
        </div>
      </fieldset>
      <div class="form-group">
        <label class="col-md-3 control-label" for="title">free</label>
        <div class="col-md-3">
          <input name="free" type="radio" value="0">tidak Gratis</div>
        <div class="col-md-3">
          <input name="free" type="radio" value="1">Gratis</div>
      </div>
    </fieldset>
    <fieldset>
      <div class="form-group">
        <label class="col-md-3 control-label" for="title">price</label>
        <div class="col-md-3">
          <input type="text" name="price">
        </div>
      </div>
    </fieldset>
    <fieldset>
      <div class="form-group">
        <label class="col-md-3 control-label" for="title">score</label>
        <div class="col-md-3">
          <input type="text" name="score">
        </div>
      </div>
    </fieldset>
    <fieldset>
      <div class="form-group">
        <label class="col-md-3 control-label" for="title">reviews</label>
        <div class="col-md-3">
          <input type="text" name="reviews">
        </div>
      </div>
    </fieldset>
    <fieldset>
      <div class="form-group">
        <label class="col-md-3 control-label" for="title">developer</label>
        <div class="col-md-3">
          <input type="text" name="developer">
        </div>
      </div>
    </fieldset>
  </div>

```

```

78         <fieldset>
79             <div class="form-group">
80                 <label class="col-md-3 control-label" for="title">developerEmail</label>
81                 <div class="col-md-3">
82                     <input type="text" name="developerEmail">
83                 </div>
84             </div>
85         </fieldset>
86         <fieldset>
87             <div class="form-group">
88                 <label class="col-md-3 control-label" for="title">developerWebsite</label>
89                 <div class="col-md-3">
90                     <input type="text" name="developerWebsite">
91                 </div>
92             </div>
93         </fieldset>
94         <fieldset>
95             <div class="form-group">
96                 <label class="col-md-3 control-label" for="title">genre</label>
97                 <div class="col-md-3">
98                     <select name="genre">
99                         <option value="ANDROID_WEAR">ANDROID_WEAR</option>
100                         <option value="ART_AND_DESIGN">ART_AND_DESIGN</option>
101                         <option value="BEAUTY">BEAUTY</option>
102                         <option value="BOOKS_AND_REFERENCE">BOOKS_AND_REFERENCE</option>
103                         <option value="BUSINESS">BUSINESS</option>
104                         <option value="COMICS">COMICS</option>
105                         <option value="COMMUNICATION">COMMUNICATION</option>
106                         <option value="DATING">DATING</option>
107                         <option value="EDUCATION">EDUCATION</option>
108                         <option value="ENTERTAINMENT">ENTERTAINMENT</option>
109                         <option value="EVENTS">EVENTS</option>
110                         <option value="FINANCE">FINANCE</option>
111                         <option value="FOOD_AND_DRINK">FOOD_AND_DRINK</option>
112                         <option value="HEALTH_AND_FITNESS">HEALTH_AND_FITNESS</option>
113                         <option value="HOUSE_AND_HOME">HOUSE_AND_HOME</option>
114                         <option value="LIBRARIES_AND_DEMO">LIBRARIES_AND_DEMO</option>
115                         <option value="LIFESTYLE">LIFESTYLE</option>
116                         <option value="MAPS_AND_NAVIGATION">MAPS_AND_NAVIGATION</option>
117                         <option value="MEDICAL">MEDICAL</option>
118                         <option value="MUSIC_AND_AUDIO">MUSIC_AND_AUDIO</option>
119                         <option value="NEWS_AND_MAGAZINES">NEWS_AND_MAGAZINES</option>
120                         <option value="PARENTING">PARENTING</option>
121                         <option value="PERSONALIZATION">PERSONALIZATION</option>
122                         <option value="PHOTOGRAPHY">PHOTOGRAPHY</option>
123                         <option value="PRODUCTIVITY">PRODUCTIVITY</option>
124                         <option value="SHOPPING">SHOPPING</option>
125                         <option value="SOCIAL">SOCIAL</option>
126                         <option value="SPORTS">SPORTS</option>
127                         <option value="TOOLS">TOOLS</option>
128                         <option value="TRAVEL_AND_LOCAL">TRAVEL_AND_LOCAL</option>
129                         <option value="VIDEO_PLAYERS">VIDEO_PLAYERS</option>
130                         <option value="WEATHER">WEATHER</option>
131                         <option value="APP_WIDGETS">APP_WIDGETS</option>
132                         <option value="GAME">GAME</option>
133                         <option value="GAME_ACTION">GAME_ACTION</option>
134                         <option value="GAME_ADVENTURE">GAME_ADVENTURE</option>
135                         <option value="GAME_ARCADE">GAME_ARCADE</option>
136                         <option value="GAME_BOARD">GAME_BOARD</option>
137                         <option value="GAME_CARD">GAME_CARD</option>
138                         <option value="GAME_CASUAL">GAME_CASUAL</option>
139                         <option value="GAME_CASUAL">GAME_CASUAL</option>
140                         <option value="GAME_CASUAL">GAME_CASUAL</option>
141                         <option value="GAME_EDUCATIONAL">GAME_EDUCATIONAL</option>
142                         <option value="GAME_MUSIC">GAME_MUSIC</option>
143                         <option value="GAME_PUZZLE">GAME_PUZZLE</option>
144                         <option value="GAME_RACING">GAME_RACING</option>
145                         <option value="GAME_ROLE_PLAYING">GAME_ROLE_PLAYING</option>
146                         <option value="GAME_SIMULATION">GAME_SIMULATION</option>
147                         <option value="GAME_SPORTS">GAME_SPORTS</option>
148                         <option value="GAME_STRATEGY">GAME_STRATEGY</option>
149                         <option value="GAME_TRIVIA">GAME_TRIVIA</option>
150                         <option value="GAME_WORD">GAME_WORD</option>
151                         <option value="FAMILY">FAMILY</option>
152                         <option value="FAMILY_ACTION">FAMILY_ACTION</option>
153                         <option value="FAMILY_BRAINGAMES">FAMILY_BRAINGAMES</option>
154                         <option value="FAMILY_CREATES">FAMILY_CREATES</option>
155                         <option value="FAMILY_EDUCATION">FAMILY_EDUCATION</option>
156                         <option value="FAMILY_MUSICVIDEO">FAMILY_MUSICVIDEO</option>

```

```

155 <option value="FAMILY_PRETEND">FAMILY_PRETEND</option>
156 </select>
157 </div>
158 </div>
159 </fieldset>
160 </div>
161 </fieldset>
162 </div>
163 <div class="form-group">
164 <label class="col-md-3 control-label" for="title">description</label>
165 <div class="col-md-3">
166 <input type="text" name="description">
167 </div>
168 </div>
169 </fieldset>
170 </div>
171 <div class="form-group">
172 <label class="col-md-3 control-label" for="title">adsupported</label>
173 <div class="col-md-3">
174 <input name="adsupported" type="radio" value="0">Tidak ada iklan</div>
175 <div class="col-md-3"> <input name="adsupported" type="radio" value="1">ada iklan</div>
176 </div>
177 </div>
178 </div>
179 </fieldset>
180 </div>
181 <div class="form-group">
182 <label class="col-md-3 control-label" for="title">contentrating</label>
183 <div class="col-md-3">
184 <input type="text" name="contentrating">
185 </div>
186 </div>
187 </fieldset>
188 </div>
189 <div class="form-group">
190 <label class="col-md-3 control-label" for="title">offersIAP</label>
191 <div class="col-md-3">
192 <input name="offersIAP" type="radio" value="0">Tidak memiliki early access</div>
193 <div class="col-md-3"> <input name="offersIAP" type="radio" value="1">ada early access</div>
194 </div>
195 </div>
196 </div>
197 </fieldset>
198 </div>
199 <div class="form-group">
200 <label class="col-md-3 control-label" for="title">video</label>
201 <div class="col-md-3">
202 <input type="text" name="video">
203 </div>
204 </div>
205 </fieldset>
206 </div>
207 <div class="form-group">
208 <label class="col-md-3 control-label" for="title">url</label>
209 <div class="col-md-3">
210 <input type="text" name="url" placeholder="url aplikasi ">
211 </div>
212 </div>
213 </fieldset>
214 </div>
215 <div class="form-group">
216 <input type="submit" name="submit" value="submit">
217 </div>
218 <{{ csrf_field() }}>
219 </form>

```

Gambar D-4 Memasukkan data aplikasi secara utuh

D - 6 -

```

@extends('layouts.main')

@section('content')

<div class="about">
<div class="container">
<center>
<h1>Insert Data App Id </h1>
</center>
<div class="clearfix"> </div>
<br>
<div>
<div class="col-md-6">
<form class="form-horizontal" action ="/listings" method="post">
<fieldset>
<div class="form-group">
<label class="col-md-8 control-label" for="title">App ID ( Masukkan app id jika ada, cth : com.netflix.mediaclient)</label>
<div class="col-md-3">
<input type="text" name="appid">
</div>
</div>
</div>
</fieldset>

<div class="form-group">
<input type="submit" name="submit" value="submit">
</div>
</div>
{{ csrf_field() }}
</form>

```

Gambar D-5 Memasukkan data aplikasi berdasarkan appId

```

<?php

namespace PlayAppSimilar\Http\Controllers;

use Illuminate\Http\Request;
use PlayAppSimilar\Model\playappsimilar;
use PlayAppSimilar\Model\playapplication;
use PlayAppSimilar\Model\appstoreapplication;

class HomeController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('auth');
    }

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        //SELECT * FROM 'playapplications' ORDER BY 'maxInstalls' DESC, score desc limit 100 SELECT * FROM 'appst
        $playtop10 = playapplication::orderBy('maxInstalls', 'desc')->take(10)->get();
        $appstoretop10 = appstoreapplication::orderBy('reviews', 'desc')->orderBy('score', 'desc')->take(10)->get();
        // join('playapplications', 'playappsimilars.appid_play', '=',->take(10)->get() 'playapplications.appid')
        // dd($playapp);
        // return view('core/home',compact('playapp'));
        return view('home', array(
            'playtop10' => $playtop10,
            'appstoretop10' => $appstoretop10
        ));
    }
}

```

```

39     }
40     public function create()
41     {
42         return view('core/create');
43     }
44     public function store(Request $request)
45     {
46         $playapplications = new playapplication;
47         $playapplications->title = $request->title;
48         $playapplications->summary = $request->summary;
49         $playapplications->icon = $request->icon;
50         $playapplications->free = $request->free;
51         $playapplications->price = $request->price;
52         $playapplications->mininstalls = '100000';
53         $playapplications->maxinstalls = '100000';
54         $playapplications->score = $request->score;
55         $playapplications->reviews = $request->reviews;
56         $playapplications->developer = $request->developer;
57         $playapplications->developerEmail = $request->developerEmail;
58         $playapplications->developerWebsite = $request->developerWebsite;
59         $playapplications->updated = 'May 10, 2017';
60         $playapplications->genre = $request->genre;
61         $playapplications->genreid = strtoupper($request->genre);
62         $playapplications->description = $request->description;
63         $playapplications->descriptionHTML = $request->description;
64         $playapplications->adssupported = $request->adssupported;
65         $playapplications->contentRating = $request->contentRating;
66         $playapplications->offersIAP = $request->offersIAP;
67         $playapplications->video = $request->video;
68         $playapplications->url = $request->url;
69         $playapplications->androidVersionText = $request->androidVersionText;
70         $playapplications->androidVersion = '4.3 ';
71         $playapplications->recentChanges = $request->recentChanges;
72         $playapplications->preregister = 0;
73         $playapplications->status_is1 = '0';
74         $playapplications->appid = 'appid';
75         $playapplications->id_sin_play = 0;
76         $playapplications->save();
77     return view('core/about');
78     }
79     public function createapp()
80     {
81         return view('core/createappid');
82     }
83     public function storeapp(Request $request)
84     {
85         $a = $request->appid;
86         $service_url = "https://googleplayapi.herokuapp.com/app/$a";
87         $curl = curl_init($service_url);
88         curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
89         curl_setopt($curl, CURLOPT_SSL_VERIFYHOST, 0);
90         curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);
91         //execute the session
92         $curl_response = curl_exec($curl);
93         //finish off the session
94         curl_close($curl);
95         $vali = json_decode($curl_response, true);
96
97         if(isset($vali['url'])){
98             $url = htmlspecialchars($vali['url'], ENT_QUOTES);
99         }
100         else{
101             $url = "";
102         }
103         if(isset($vali['title'])){
104             $title = htmlspecialchars($vali['title'], ENT_QUOTES);
105         }
106         else{
107             $title = "";
108         }
109         if(isset($vali['appid'])){
110             $appid = htmlspecialchars($vali['appid'], ENT_QUOTES);
111         }
112         else{
113             $appid = "";
114         }

```

```

115     if(isset($vali['summary'])){
116         $summary = htmlspecialchars($vali['summary'], ENT_QUOTES);
117     }
118     else{
119         $summary = "";
120     }
121     if(isset($vali['developer'])){
122         $developer = htmlspecialchars($vali['developer'], ENT_QUOTES);
123     }
124     else{
125         $developer = "";
126     }
127     if(isset($vali['icon'])){
128         $icon = htmlspecialchars($vali['icon'], ENT_QUOTES);
129     }
130     else{
131         $icon = "";
132     }
133     if(isset($vali['price'])){
134         $price = htmlspecialchars($vali['price'], ENT_QUOTES);
135     }
136     else{
137         $price = "";
138     }
139     if(isset($vali['free'])){
140         $free = htmlspecialchars($vali['free'], ENT_QUOTES);
141     }
142     else{
143         $free = "";
144     }
145     if(isset($vali['score'])){
146         $score = htmlspecialchars($vali['score'], ENT_QUOTES);
147     }
148     else{
149         $score = "";
150     }
151     if(isset($vali['minInstalls'])){
152         $minInstalls = htmlspecialchars($vali['minInstalls'], ENT_QUOTES);
153     }
154     else{
155         $minInstalls = "";
156     }
157     // $minInstalls = htmlspecialchars($vali['minInstalls'], ENT_QUOTES);
158     if(isset($vali['maxInstalls'])){
159         $maxInstalls = htmlspecialchars($vali['maxInstalls'], ENT_QUOTES);
160     }
161     else{
162         $maxInstalls = "";
163     }
164     // $maxInstalls = htmlspecialchars($vali['maxInstalls'], ENT_QUOTES);
165     if(isset($vali['reviews'])){
166         $reviews = htmlspecialchars($vali['reviews'], ENT_QUOTES);
167     }
168     else{
169         $reviews = "";
170     }
171     // $reviews = htmlspecialchars($vali['reviews'], ENT_QUOTES);
172     if(isset($vali['developerEmail'])){
173         $developerEmail = htmlspecialchars($vali['developerEmail'], ENT_QUOTES);
174     }
175     else{
176         $developerEmail = "";
177     }
178     // $developerEmail = htmlspecialchars($vali['developerEmail'], ENT_QUOTES);
179     if(isset($vali['developerWebsite'])){
180         $developerWebsite = htmlspecialchars($vali['developerWebsite'], ENT_QUOTES);
181     }
182     else{
183         $developerWebsite = "";
184     }
185     if(isset($vali['updated'])){
186         $updated = htmlspecialchars($vali['updated'], ENT_QUOTES);
187     }

```

```

193     }
194     else{
195         $updated = "";
196     }
197     if(isset($vali['genre'])){
198         $genre = htmlspecialchars($vali['genre'], ENT_QUOTES);
199     }
200     }
201     else{
202         $genre = "";
203     }
204     if(isset($vali['genreid'])){
205         $genreid = htmlspecialchars($vali['genreid'], ENT_QUOTES);
206     }
207     else{
208         $genreid = "";
209     }
210     if(isset($vali['description'])){
211         $description = htmlspecialchars($vali['description'], ENT_QUOTES);
212     }
213     }
214     else{
215         $description = "";
216     }
217     if(isset($vali['descriptionHTML'])){
218         $descriptionHTML = htmlspecialchars($vali['descriptionHTML'], ENT_QUOTES);
219     }
220     }
221     else{
222         $descriptionHTML = "";
223     }
224 }
225 if(isset($vali['offersIAP']) || empty($vali['offersIAP'])){
226     $offersIAP = '0';
227 }
228 else{
229     $offersIAP = htmlspecialchars($vali['offersIAP'], ENT_QUOTES);
230 }
231
232 if(isset($vali['androidVersionText'])){
233     $androidVersionText = htmlspecialchars($vali['androidVersionText'], ENT_QUOTES);
234 }
235 else{
236     $androidVersionText = "";
237 }
238 if(isset($vali['androidVersion'])){
239     $androidVersion = htmlspecialchars($vali['androidVersion'], ENT_QUOTES);
240 }
241 else{
242     $androidVersion = "";
243 }
244 if(isset($vali['contentRating'])){
245     $contentRating = htmlspecialchars($vali['contentRating'], ENT_QUOTES);
246 }
247 else{
248     $contentRating = "";
249 }
250 if(isset($vali['video'])){
251     $video = htmlspecialchars($vali['video'], ENT_QUOTES);
252 }
253 else{
254     $video = "";
255 }
256 if(isset($vali['recentChanges'])){
257     $recentChanges = htmlspecialchars(implode(", ", $vali['recentChanges']), ENT_QUOTES);
258 }
259 else if(strlen($vali['recentChanges'])==0){
260     $recentChanges = "";
261 }
262 else {
263     $recentChanges = htmlspecialchars($vali['recentChanges'], ENT_QUOTES);
264 }
265 if(isset($vali['preregister']) || empty($vali['preregister'])){
266     $preregister = '0';
267 }
268 else{

```

```

277     $preregister = htmlspecialchars($val['preregister'], ENT_QUOTES);
278 }
279 $playapplications = new playapplication;
280 $playapplications->title = $title;
281 $playapplications->summary = $summary;
282 $playapplications->icon = $icon;
283 $playapplications->free = $free;
284 $playapplications->price = $price;
285 $playapplications->minInstalls = $minInstalls;
286 $playapplications->maxInstalls = $maxInstalls;
287 $playapplications->score = $score;
288 $playapplications->reviews = $reviews;
289 $playapplications->developer = $developer;
290 $playapplications->developerEmail = $developerEmail;
291 $playapplications->developerWebsite = $developerWebsite;
292 $playapplications->updated = $updated;
293 $playapplications->genre = $genre;
294 $playapplications->genreId = $genreId;
295 $playapplications->description = $description;
296 $playapplications->descriptionHTML = $descriptionHTML;
297 $playapplications->adSupported = $adSupported;
298 $playapplications->contentRating = $contentRating;
299 $playapplications->offersIAP = $offersIAP;
300 $playapplications->video = $video;
301 $playapplications->url = $url;
302 $playapplications->androidVersionText = $androidVersionText;
303 $playapplications->androidVersion = $androidVersion;
304 $playapplications->recentChanges = $recentChanges;
305 $playapplications->preregister = $preregister;
306 $playapplications->status_isi = '0';
307 $playapplications->appId = $a;
308 $playapplications->status_isi = '0';
309 $playapplications->id_sim_play = '0';
310 // dd($adSupported);
311 $playapplications->save();
312 return view('core/about');
313 }
314 }

```

Gambar D-6 Controller HomeController mengatur masukan data ke basis data


```

1 <?php
2
3 namespace PlayAppSimilar\Http\Controllers\Auth;
4
5 use PlayAppSimilar\Http\Controllers\Controller;
6 use Illuminate\Foundation\Auth\AuthenticatesUsers;
7
8 class LoginController extends Controller
9 {
10     /**
11      * -----
12      * Login Controller
13      * -----
14      *
15      * This controller handles authenticating users for the application and
16      * redirecting them to your home screen. The controller uses a trait
17      * to conveniently provide its functionality to your applications.
18      *
19      */
20
21     use AuthenticatesUsers;
22
23     /**
24      * Where to redirect users after login.
25      *
26      * @var string
27      */
28     protected $redirectTo = '/listing';
29
30     /**
31      * Create a new controller instance.
32      *
33      * @return void
34      */
35     public function __construct()
36     {
37         $this->middleware('guest')->except('logout');
38     }
39 }

```

Gambar D-7 Controller yang mengatur proses login

```

<?php

namespace PlayAppSimilar\Http\Controllers\Auth;

use PlayAppSimilar\User;
use PlayAppSimilar\Http\Controllers\Controller;
use Illuminate\Support\Facades\Validator;
use Illuminate\Foundation\Auth\RegistersUsers;

class RegisterController extends Controller
{
    use RegistersUsers;

    protected $redirectTo = '/listing';

    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('guest');
    }

    /**
     * Get a validator for an incoming registration request.
     *
     * @param array $data
     * @return \Illuminate\Contracts\Validation\Validator
     */
    protected function validator(array $data)
    {
        return Validator::make($data, [
            'name' => 'required|string|max:255',
            'email' => 'required|string|email|max:255|unique:users',
            'password' => 'required|string|min:6|confirmed',
        ]);
    }

    /**
     * Create a new user instance after a valid registration.
     *
     * @param array $data
     * @return User
     */
    protected function create(array $data)
    {
        return User::create([
            'name' => $data['name'],
            'email' => $data['email'],
            'password' => bcrypt($data['password']),
        ]);
    }
}

```

Gambar D-8 Controller yang mengatur proses pendaftaran

```

1 @extends('layouts.app')

2
3 @section('content')
4 <div class="container">
5   <div class="row">
6     <div class="col-md-8 col-md-offset-2">
7       <div class="panel panel-default">
8         <div class="panel-heading">login</div>
9         <div class="panel-body">
10           <form class="form-horizontal" roles="form" method="POST" action="{{ route('login') }}">
11             {{ csrf_field() }}
12
13             <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
14               <label for="email" class="col-md-4 control-label">E-Mail Address</label>
15
16               <div class="col-md-6">
17                 <input id="email" type="email" class="form-control" name="email" value="{{ old('email') }}" required autofocus>
18
19                 @if ($errors->has('email'))
20                   <span class="help-block">
21                     <strong>{{ $errors->first('email') }}</strong>
22                   </span>
23                 @endif
24               </div>
25             </div>
26
27             <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">
28               <label for="password" class="col-md-4 control-label">Password</label>
29
30               <div class="col-md-6">
31                 <input id="password" type="password" class="form-control" name="password" required>
32
33                 @if ($errors->has('password'))
34                   <span class="help-block">
35                     <strong>{{ $errors->first('password') }}</strong>
36                   </span>
37                 @endif
38               </div>
39             </div>
40
41             <div class="form-group">
42               <div class="col-md-8 col-md-offset-4">
43                 <div class="checkbox">
44                   <label>
45                     <input type="checkbox" name="remember" {{ old('remember') ? 'checked' : '' }}> Remember Me
46                   </label>
47                 </div>
48               </div>
49
50               <div class="form-group">
51                 <div class="col-md-8 col-md-offset-4">
52                   <button type="submit" class="btn btn-primary">
53                     Login
54                   </button>
55                   <a class="btn btn-primary" href="{{ route('register') }}">
56                     Register
57                   </a>
58                   <a class="btn btn-link" href="{{ route('password.request') }}">
59                     Forgot Your Password?
60                   </a>
61                 </div>
62               </div>
63             </form>
64           </div>
65         </div>
66       </div>
67     </div>
68   </div>
69 </div>
70 @endsection

```

Gambar D-9 Kode untuk menampilkan kotak login

```

1 @extends('layouts.app')

2
3 @section('content')
4 <div class="container">
5   <div class="row">
6     <div class="col-md-8 col-md-offset-2">
7       <div class="panel panel-default">
8         <div class="panel-heading">Register</div>
9         <div class="panel-body">
10          <form class="form-horizontal" role="form" method="POST" action="{{ route('register') }}">
11            {{ csrf_field() }}
12
13            <div class="form-group{{ $errors->has('name') ? ' has-error' : '' }}">
14              <label for="name" class="col-md-4 control-label">Name</label>
15
16              <div class="col-md-6">
17                <input id="name" type="text" class="form-control" name="name" value="{{ old('name') }}" required autofocus>
18
19                @if ($errors->has('name'))
20                  <span class="help-block">
21                    <strong>{{ $errors->first('name') }}</strong>
22                  </span>
23                @endif
24              </div>
25            </div>
26
27            <div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
28              <label for="email" class="col-md-4 control-label">E-Mail Address</label>
29
30              <div class="col-md-6">
31                <input id="email" type="email" class="form-control" name="email" value="{{ old('email') }}" required>
32
33                @if ($errors->has('email'))
34                  <span class="help-block">
35                    <strong>{{ $errors->first('email') }}</strong>
36                  </span>
37                @endif
38              </div>
39            </div>
40
41            <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}">
42              <label for="password" class="col-md-4 control-label">Password</label>
43
44              <div class="col-md-6">
45                <input id="password" type="password" class="form-control" name="password" required>
46
47                @if ($errors->has('password'))
48                  <span class="help-block">
49                    <strong>{{ $errors->first('password') }}</strong>
50                  </span>
51                @endif
52              </div>
53            </div>
54
55            <div class="form-group">
56              <label for="password-confirm" class="col-md-4 control-label">Confirm Password</label>
57
58              <div class="col-md-6">
59                <input id="password-confirm" type="password" class="form-control" name="password_confirmation" required>
60              </div>
61            </div>
62
63            <div class="form-group">
64              <div class="col-md-6 col-md-offset-4">
65                <button type="submit" class="btn btn-primary">
66                  Register
67                </button>
68              </div>
69            </div>
70          </form>
71        </div>
72      </div>
73    </div>
74  </div>
75 @endsection

```

Gambar D-10 Kode untuk menampilkan kotak pendaftaran

```

1  @extends('layouts.main')
2
3  @section('content')
4      <div class="mobiles">
5          <div class="container">
6              <div class="w3ls_mobiles_grids">
7                  <div class="clearfix"> </div>
8                  @foreach ($reviewapp as $r1)
9                      <div class="col-md-4 agileinfo_new_products_grid agileinfo_new_products_grid_mobiles">
10                         <div class="agile_ecommerce_tab_left mobiles_grid">
11                             <div>
12                                 
13                             </div>
14                             <p>User Pengguna : {{ $r1->userName }}</p>
15                             <p>Tanggal : {{ $r1->date }}</p>
16                             <p>Rating : {{ $r1->score }}</p>
17                             <p>Review : {{ $r1->text }}</p>
18                         </div>
19                     </div>
20                 @endforeach
21             </div>
22             <div class="clearfix"> </div>
23         </div>
24     </div>
25 </div>
26 <center>
27     {{ $reviewapp->links() }}
28 </center>
29 @endsection

```

Gambar D-11 Kode untuk menampilkan ulasan aplikasi

```

public function review($id){
    $reviewapp = review::where('app_id','=', $id)->paginate(12);
    return view('core/review',compact('reviewapp'));
}

```

Gambar D-12 Controller yang mengatur tampilan ulasan aplikasi

```

1  <?php
2
3  namespace PlayAppSimilar\Model;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class review extends Model
8  {
9      public $timestamps = false;
10      //
11  }

```

Gambar D-13 Model untuk memanggil tabel reviews

D - 16 -

(Halaman ini sengaja dikosongkan)

Lampiran E

Lembar Persetujuan

Pendahuluan

Testing ini dilakukan untuk melakukan penilaian terhadap kegunaan (Usabilitas) dari aplikasi yang akan dibuat oleh peneliti.

Aplikasi yang akan dibuat diberi nama PlayAppSimilar. PlayAppSimilar merupakan prototype aplikasi yang menghadirkan fitur seperti daftar aplikasi yang tersedia di play store dan app store, selain itu juga menyuguhkan top aplikasi berdasarkan jumlah downloadnya..

Pengembangan aplikasi PlayAppSimilar memiliki tujuan yaitu untuk membantu pengguna dalam melihat aplikasi apa saja yang tersedia di app store dan play store dan membantu developer memasukkan karyanya.

Usability testing ini akan memakan waktu sekitar 10 menit dan media yang digunakan adalah aplikasi *Minimum Viable Product*. Harapannya hasil masukan dari *testing* akan dibuat untuk mengembangkan aplikasi kedepannya. seluruh fitur dan fungsi dari aplikasi dibuat berdasarkan literatur dan wawancara dengan pihak yang terkait dengan penelitian ini.

Terdapat 2 Bagian pertanyaan yang harus dijawab. Pada bagian pertama menggunakan tipe soal likert dengan skala 1 – 5 yang bertujuan mengukur usabilitas dari aplikasi. Pada bagian kedua menggunakan tipe soal bebas, untuk mengambil evaluasi dan saran untuk pengembangan aplikasi.

Terima kasih atas partisipasi anda.

FORM PERSETUJUAN

Saya mengerti bahwa partisipasi dalam studi kegunaan ini bersifat sukarela.

Silakan isi form berikut untuk menunjukkan bahwa Anda telah membaca dan Anda memahami informasi pada formulir ini dan bahwa setiap pertanyaan yang mungkin Anda miliki tentang sesi telah dijawab.

Tanggal : _____
Nama Lengkap : _____
Jenis Kelamin : _____
Usia : _____

Tanda tangan : _____

Lampiran F

System Usability Scale Form

Instruksi : Isikan setiap jawaban pernyataan dengan memberikan tanda checklist (√) atau cross (x) pada salah satu kolom 1 – 5 yang menurut Anda paling sesuai dengan pendapat Anda.

1 = Sangat tidak setuju

5 = Sangat setuju

Pernyataan	1	2	3	4	5
Saya berpikir bahwa saya ingin menggunakan sistem ini sesering mungkin					
Saya berpikir sistem ini sangat rumit dan tidak perlu					
Saya berpikir bahwa sistem ini mudah digunakan					
Saya berpikir saya akan membutuhkan bantuan seseorang untuk dapat menggunakan sistem ini					
Saya menemukan seluruh fungsi yang telah terintegrasi dengan sangat baik					
Saya berpikir terlalu banyak inkonsistensi dari sistem ini					
Saya berpikir banyak orang yang akan dapat belajar menggunakan sistem ini dengan cepat					
Saya berpikir bahwa sistem ini sangat sulit untuk digunakan					
Saya merasa sangat percaya diri dalam menggunakan sistem ini					
Saya perlu belajar banyak hal sebelum dapat menggunakan sistem ini					

BAGIAN II EVALUASI

1. Adakah fungsi dari aplikasi yang anda rasa tidak perlu ada? sebutkan

2. Adakah fungsi dari sistem yang menurut Anda penting namun belum terdapat pada aplikasi? Sebutkan

3. Saran anda untuk perbaikan kedepannya?
